Analysis of
High-Dimensional Data

Leif Kobbelt

Visual Computing Institute

RWTH AACHEN UNIVERSITY

Motivation

- Given: n samples in d-dimensional space

$$\mathbf{X} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_n\right) \in R^{d \times n}$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Motivation

- Given: n samples in d-dimensional space

$$\mathbf{X} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_n\right) \in R^{d \times n}$$

- Decrease d $\Rightarrow$ dimensionality reduction:
  - PCA
  - MDS

Principal Component Analysis

- Idea: Compute orthorgonal linear transformation that transforms the data into a new coordinate system s.t.
  - greatest variance on first coordinate axis
  - second greatest variance on second axis
  - etc.
- Optimal transform for a given data set in the least squares sense
- Dimensionality reduction: project data into lower dimensional space spanned by first principal components

Principal Component Analysis

Given: n samples scattered in d-dimensional space, written as a matrix

$$\mathbf{X} = \left(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\right) \in R^{d \times n}$$

compute the centered covariance matrix:

$$C = (X - \overline{X})(X - \overline{X})^T \in R^{d \times d}$$

(interpretation as map from R^d to R^d)

Principal Component Analysis

computation of C with the *"centering matrix"*:

$$C = (XJ)(XJ)^T = X\,J\,J^T X^T$$

$$J = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$$

principal component(s):

eigenvector(s) $\boldsymbol{v}_i$ to largest eigenvalue(s) λ_i of C

(low rank approximation)

Principal Component Analysis

$$C = V\,D\,V^T$$

$$= \left[\mathbf{v}_1 \ldots \mathbf{v}_d\right] \mathrm{diag}\left[\lambda_1 \ldots \lambda_d\right] \left[\mathbf{v}_1 \ldots \mathbf{v}_d\right]^T$$

$$\approx \left[\mathbf{v}_1 \ldots \mathbf{v}_q\right] \mathrm{diag}\left[\lambda_1 \ldots \lambda_q\right] \left[\mathbf{v}_1 \ldots \mathbf{v}_q\right]^T$$

$$X^* := \left[\mathbf{v}_1 \ldots \mathbf{v}_q\right]^T X\,J \quad \in R^{q \times n}$$

Relation to SVD

- singular value decomposition

$$XJ = V \Sigma U^T$$

$$C = XJ\,(XJ)^T = V \Sigma U^T U \Sigma^T V^T$$
$$= V \Sigma^2 V^T$$

… for very large dimension *d*

$$C = XJ\,(XJ)^T \quad \in R^{d\times d}$$

$$\tilde{C} = (XJ)^T\,XJ \quad \in R^{n\times n}$$

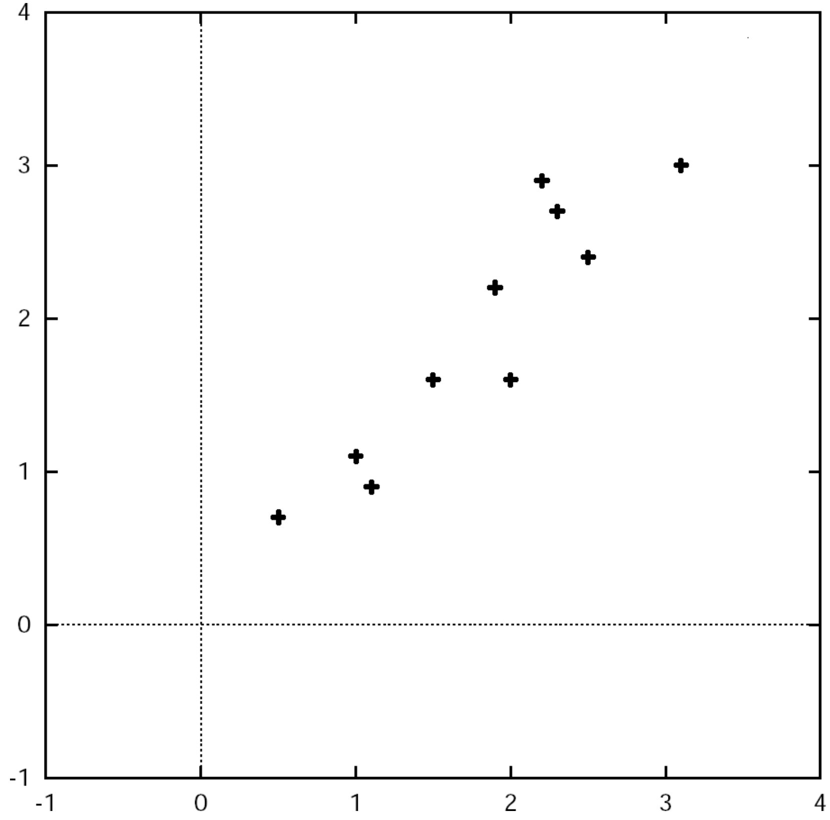
$$C\,v = \lambda\,v \qquad w = (XJ)^T\,v$$

$$\tilde{C}\,w = (XJ)^T\,XJ\,(XJ)^T\,v = \lambda\,(XJ)^T\,v = \lambda\,w$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Example

10 points in R^2



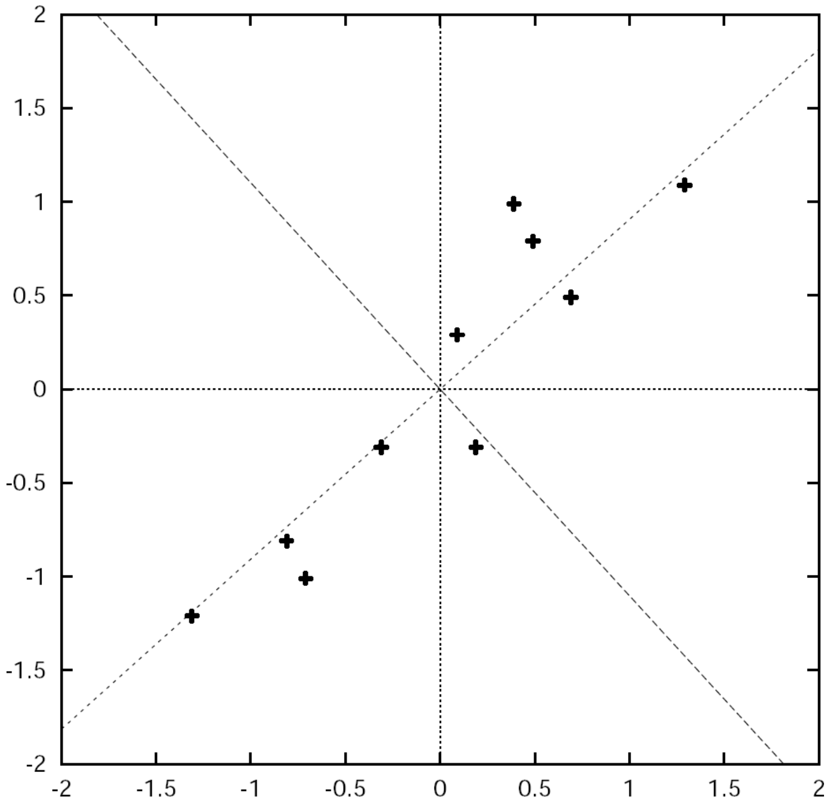
Example

10 points in R^2

$$C = \begin{pmatrix} 0.617 & 0.615 \\ 0.615 & 0.717 \end{pmatrix}$$

$$e_1 = \begin{pmatrix} -0.74 \\ 0.68 \end{pmatrix}$$

$$e_2 = \begin{pmatrix} -0.68 \\ -0.74 \end{pmatrix}$$



Multi-Dimensional Scaling

Given: For n unknown samples $\mathbf{X} \in R^{d \times n}$ in high-dimensional space

$$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \quad \mathbf{x}_i \in R^d$$

we are given a matrix $D \in R^{n \times n}$ of pairwise (squared) distances:

$$D_{i,j} = \|x_i - x_j\|^2$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Multi-Dimensional Scaling

samples $\mathbf{X}$ in some *abstract* space:

$$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \quad \mathbf{x}_i \in A$$

matrix $D \in R^{n\times n}$ of pairwise *abstract* distances:

$$D_{i,j}$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Multi-Dimensional Scaling

Goal: find an embedding of $\mathbf{X}$ in a low-dimensional space such that the pairwise (variations of) distances $\hat{D}$ are preserved.

$$\rho(D, \hat{D}) = \left\| J^T (D - \hat{D}) J \right\|_F^2$$

other measures $\rho(D, \hat{D})$ are possible but they cannot be solved easily.

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Multi-Dimensional Scaling

closed form solution:

first q eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_q$ of the matrix

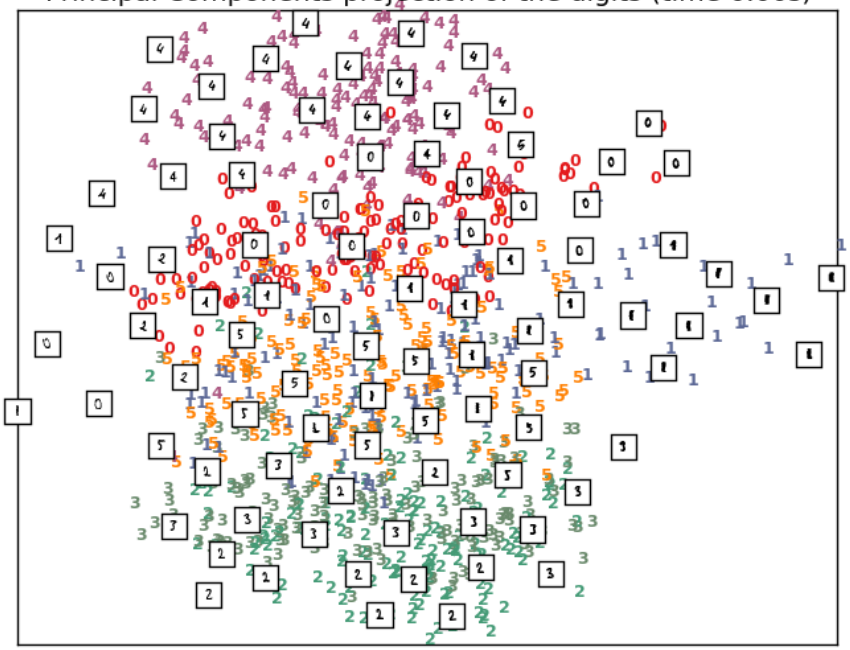
$$-\frac{1}{2} J^T D J \quad \in R^{n \times n}$$

define the coordinates of a q-dimensional embedding

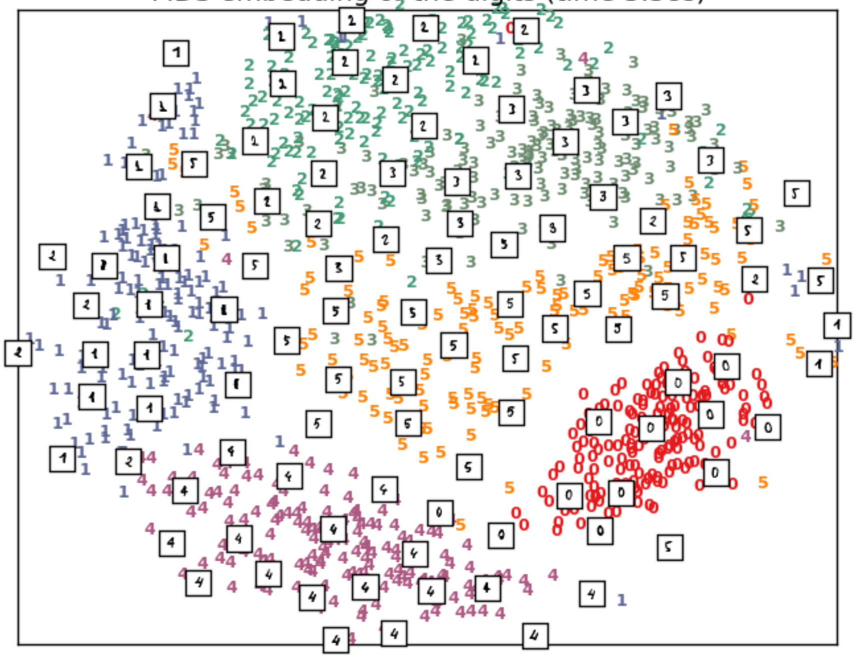
$$\mathbf{X}' = \left[\sqrt{\lambda_1} \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}, \cdots, \sqrt{\lambda_q} \frac{\mathbf{v}_q}{\|\mathbf{v}_q\|} \right]^T \quad \in R^{q \times n}$$

Multi-Dimensional Scaling

Principal Components projection of the digits (time 0.00s)



MDS embedding of the digits (time 3.38s)



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Motivation

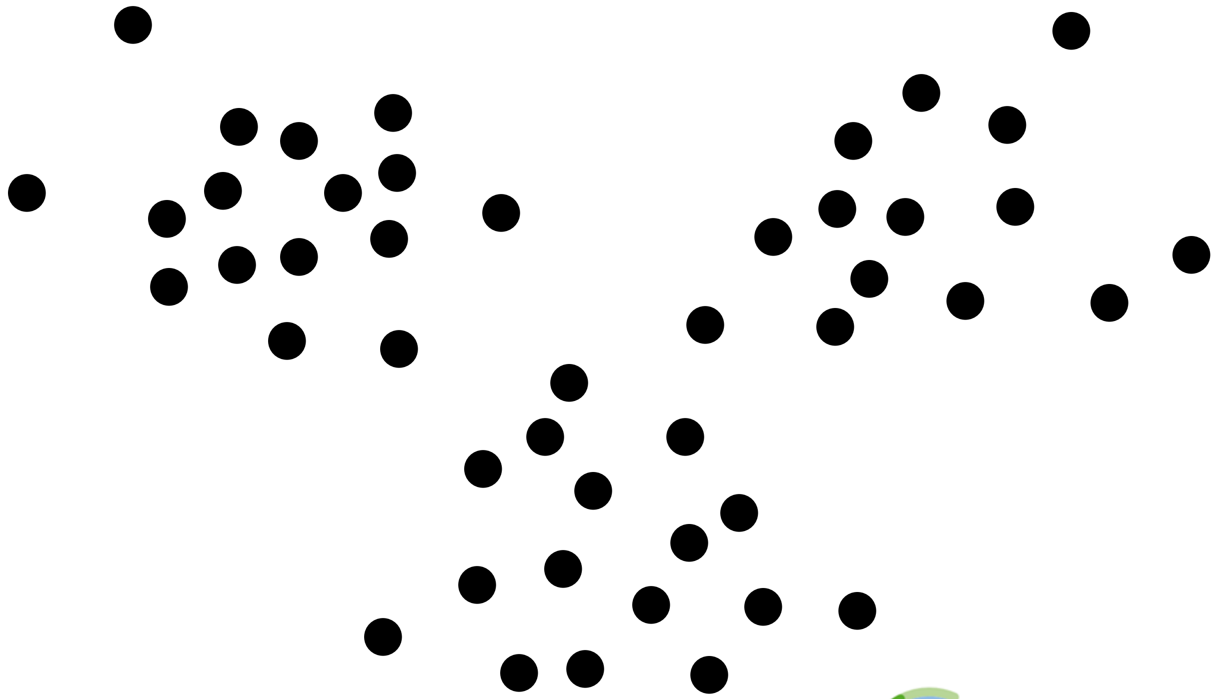
- Given: n samples in d-dimensional space

$$\mathbf{X} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_n\right) \in R^{d \times n}$$

- Decrease n ⇨ clustering:
  - k-means
  - EM
  - Mean shift
  - Spectral clustering
  - Hierarchical clustering

Cluster Analysis

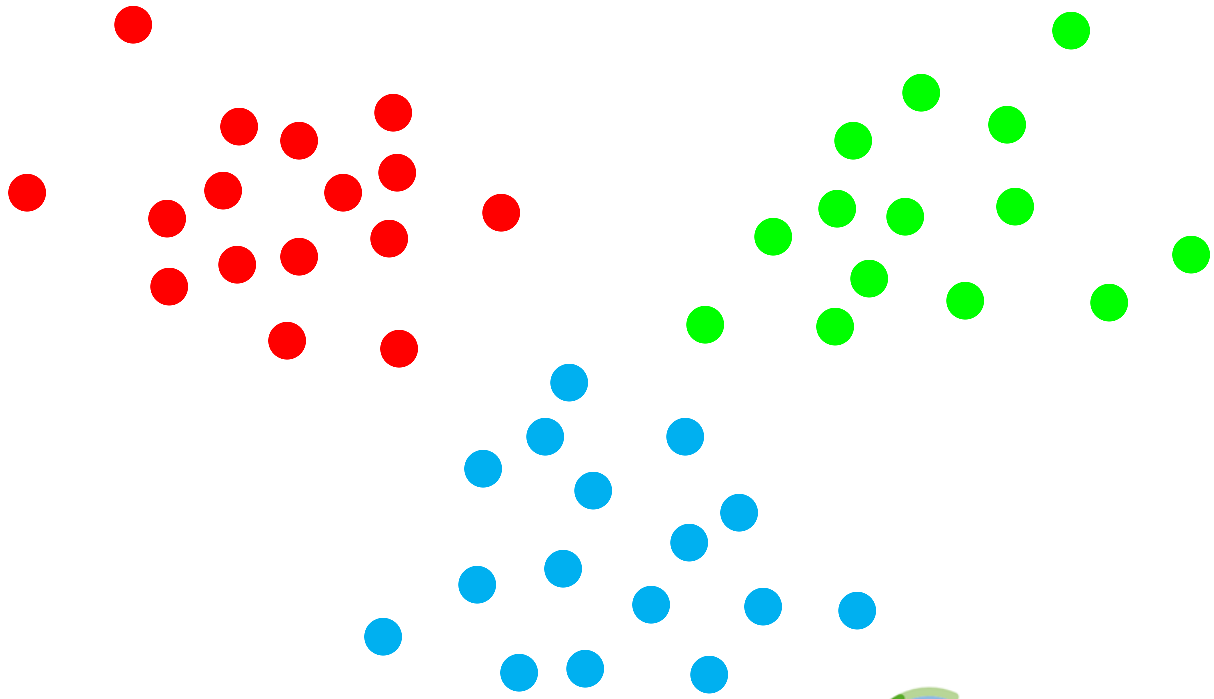
- Task: Given a set of observations / data samples, assign them into clusters so that observations in the same cluster are similar.



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Cluster Analysis

- Task: Given a set of observations / data samples, assign them into clusters so that observations in the same cluster are similar.



k-means Clustering

- Idea: partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

- Given: data samples $\mathbf{x}_1, \cdots, \mathbf{x}_n \quad \mathbf{x}_i \in R^d$
- Goal: partition the n samples into k sets ($k \leq n$) S_1, S_2, …, S_k such that

$$\arg\min_S = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$

is minimized, where μ_i is the mean of points in S_i.

k-means Clustering

- Two step algorithm:
  - Assignment step: Assign each sample to the cluster with the closest mean (Voronoi Diagram)

$$S_i^t = \left\{ \mathbf{x}_j : \left\| \mathbf{x}_j - \mathbf{m}_i^t \right\| \le \left\| \mathbf{x}_j - \mathbf{m}_{i^*}^t \right\|, \forall i^* = 1, \ldots, k \right\}$$

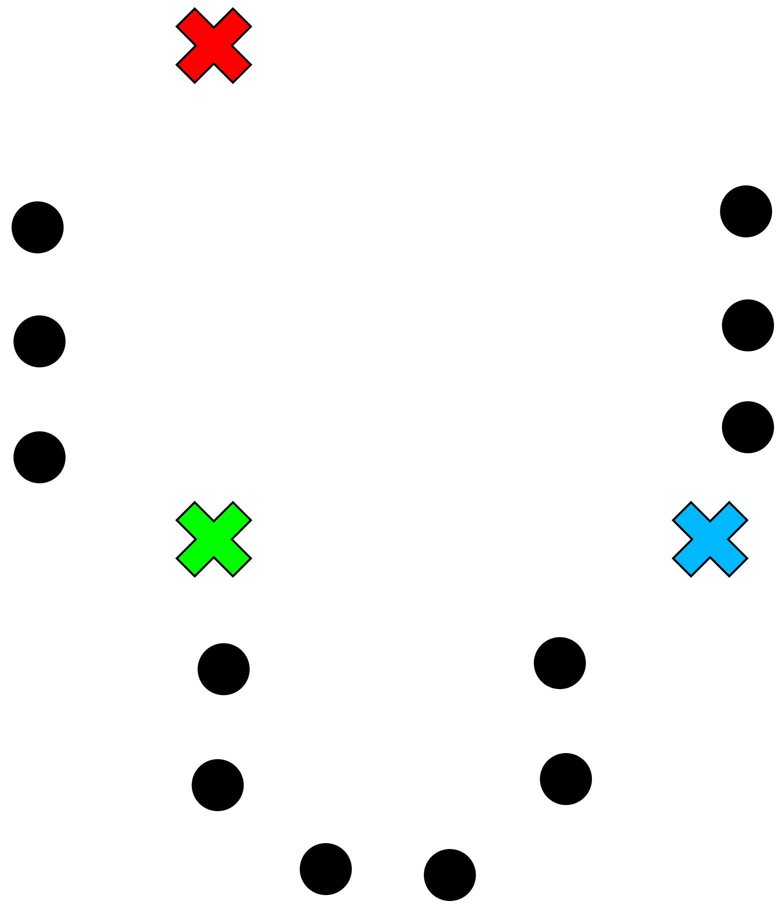
  - Update step: Calculate the new means to be the centroid of the observations in the cluster.

$$\mathbf{m}_i^{t+1} = \frac{1}{\left| S_i^t \right|} \sum_{\mathbf{x}_j \in S_i^t} \mathbf{x}_j$$

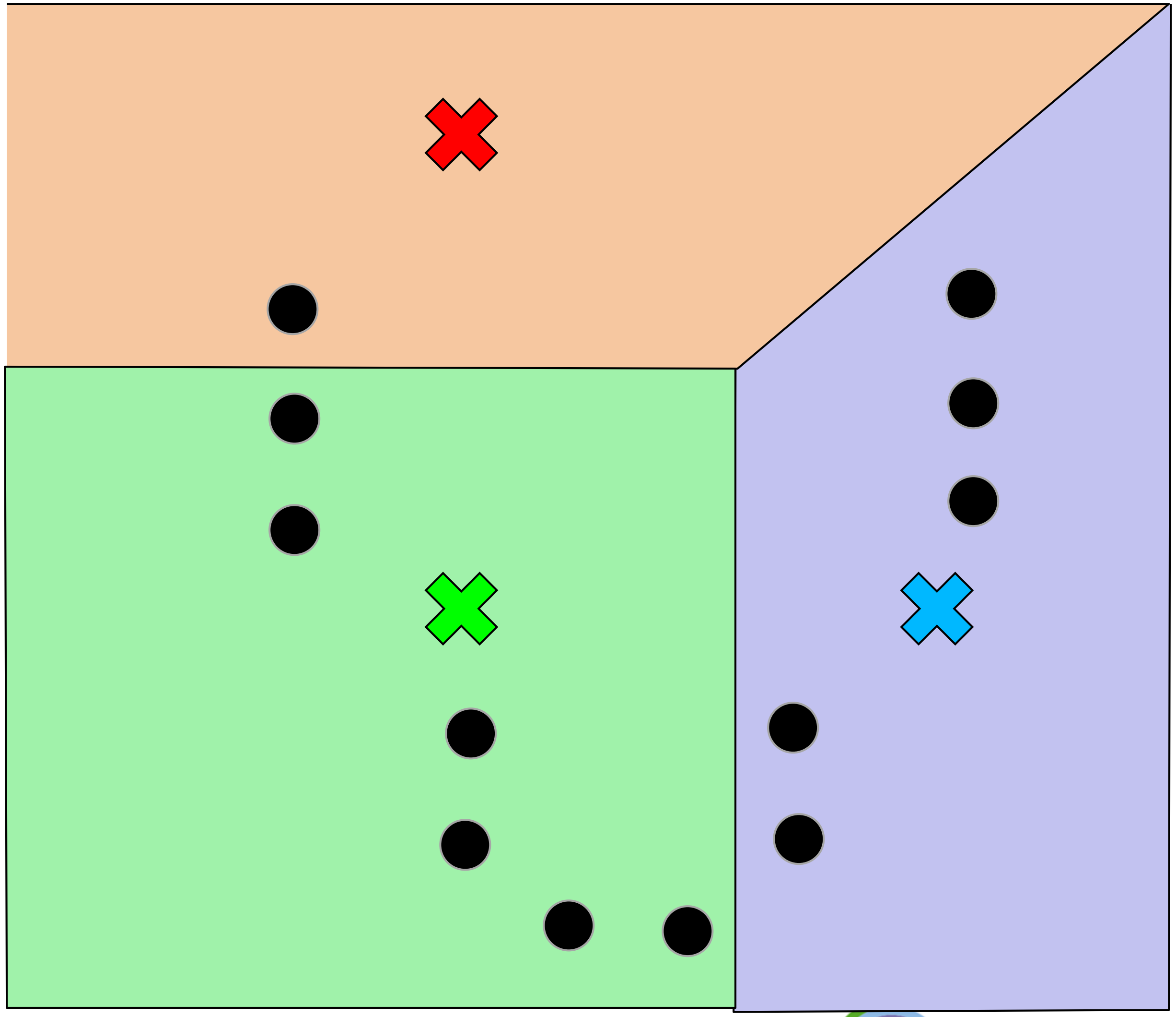
  - Iterate until convergence (assignments change no longer)

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

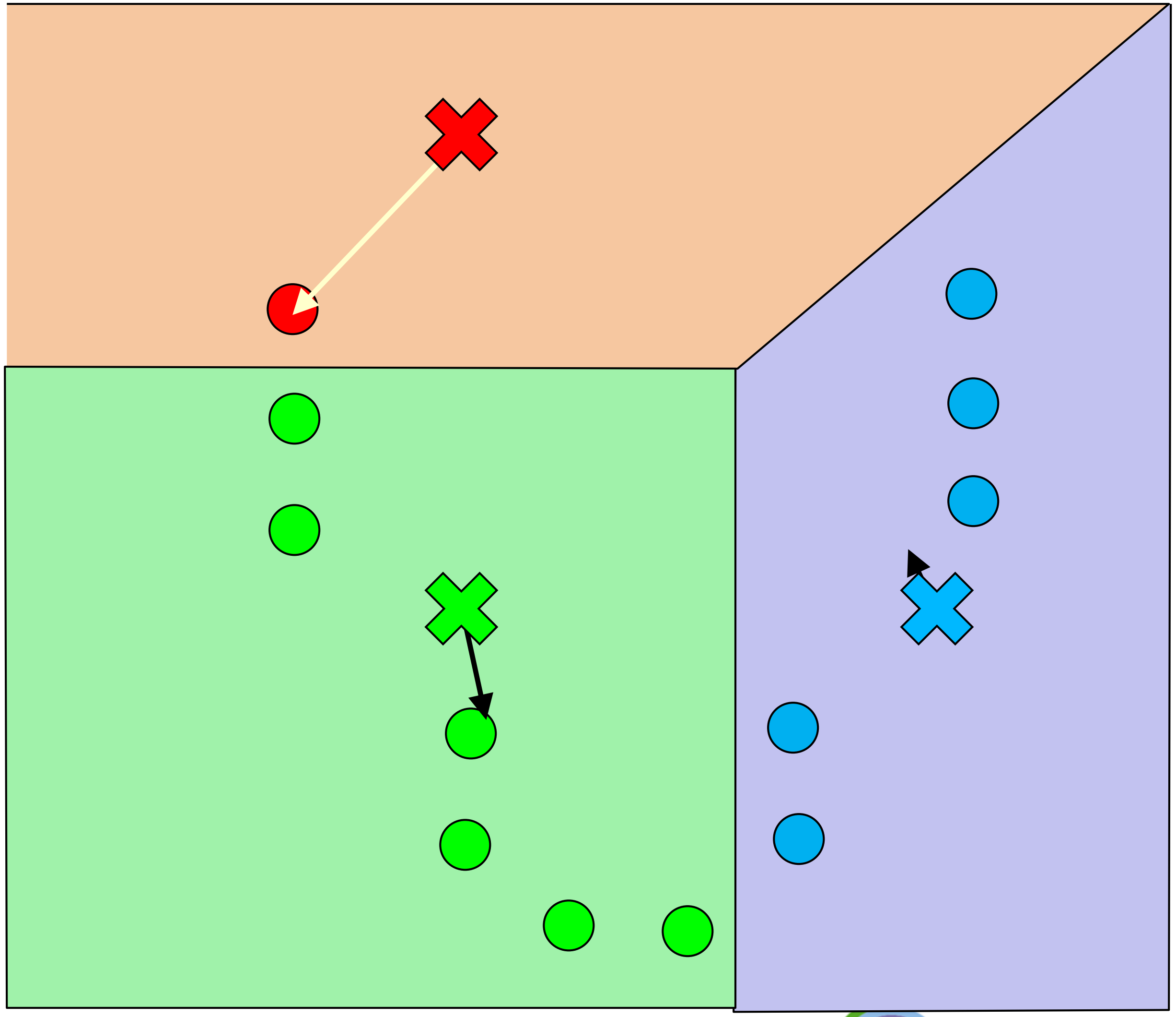
k-means Clustering



k-means Clustering

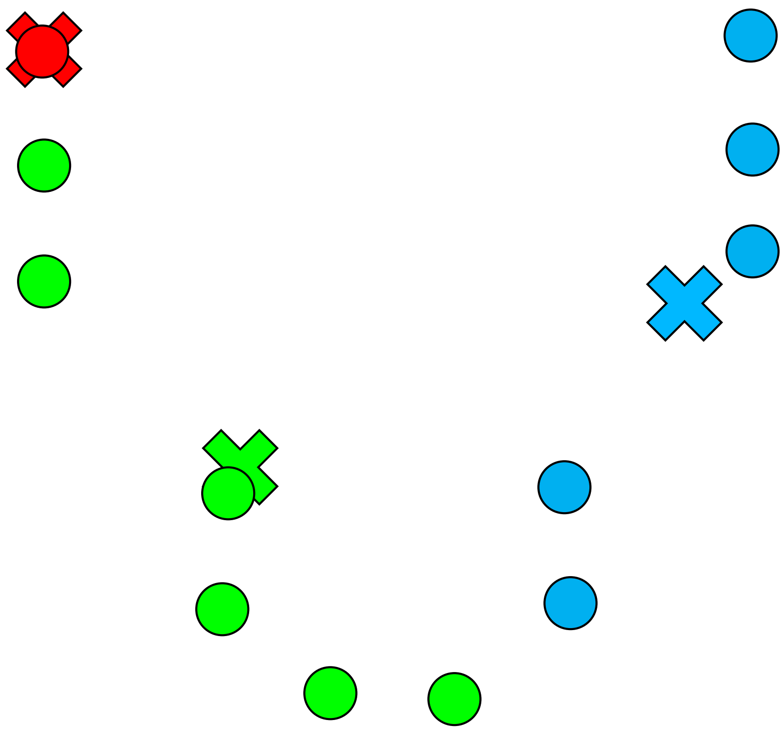


k-means Clustering

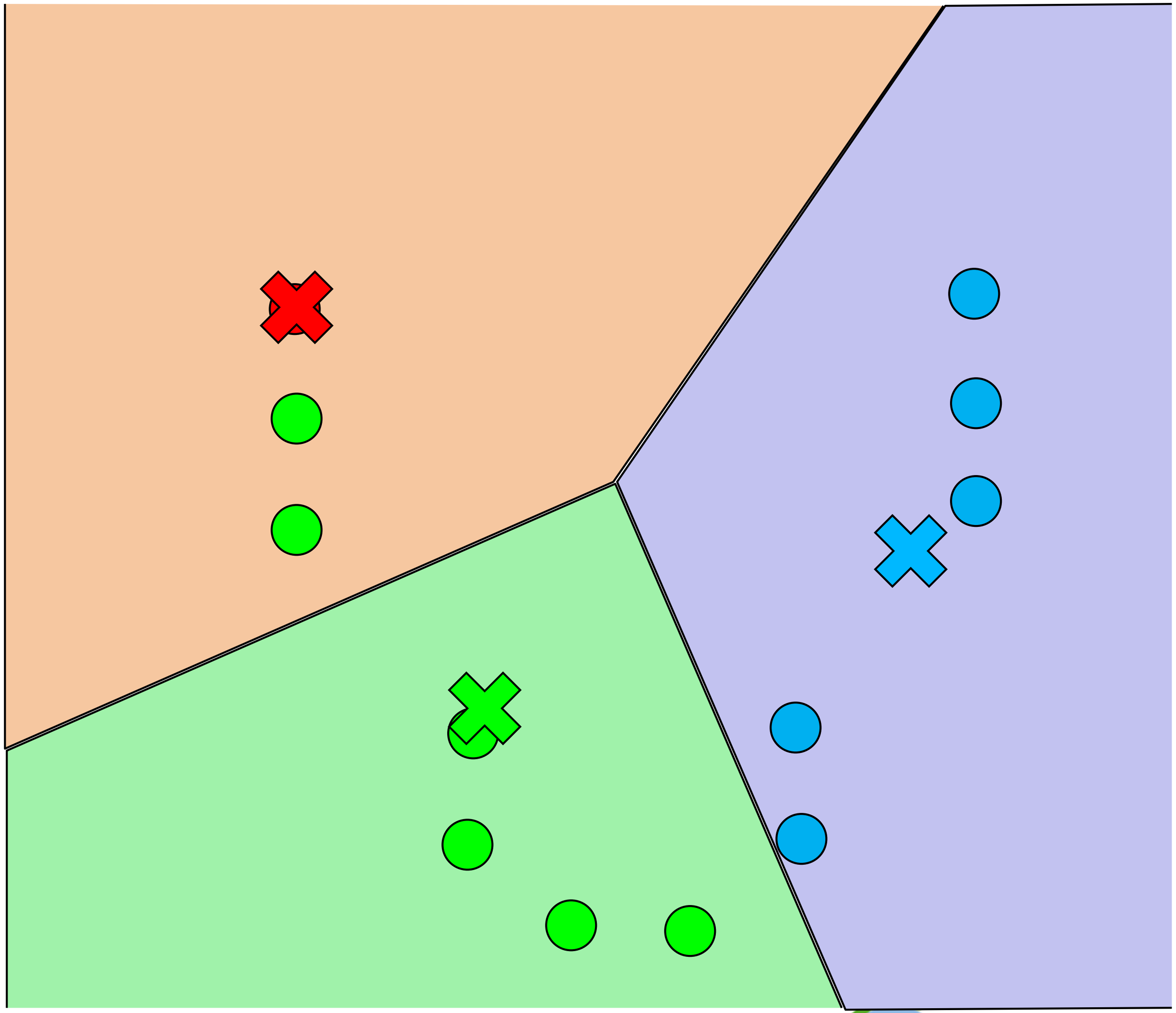


Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

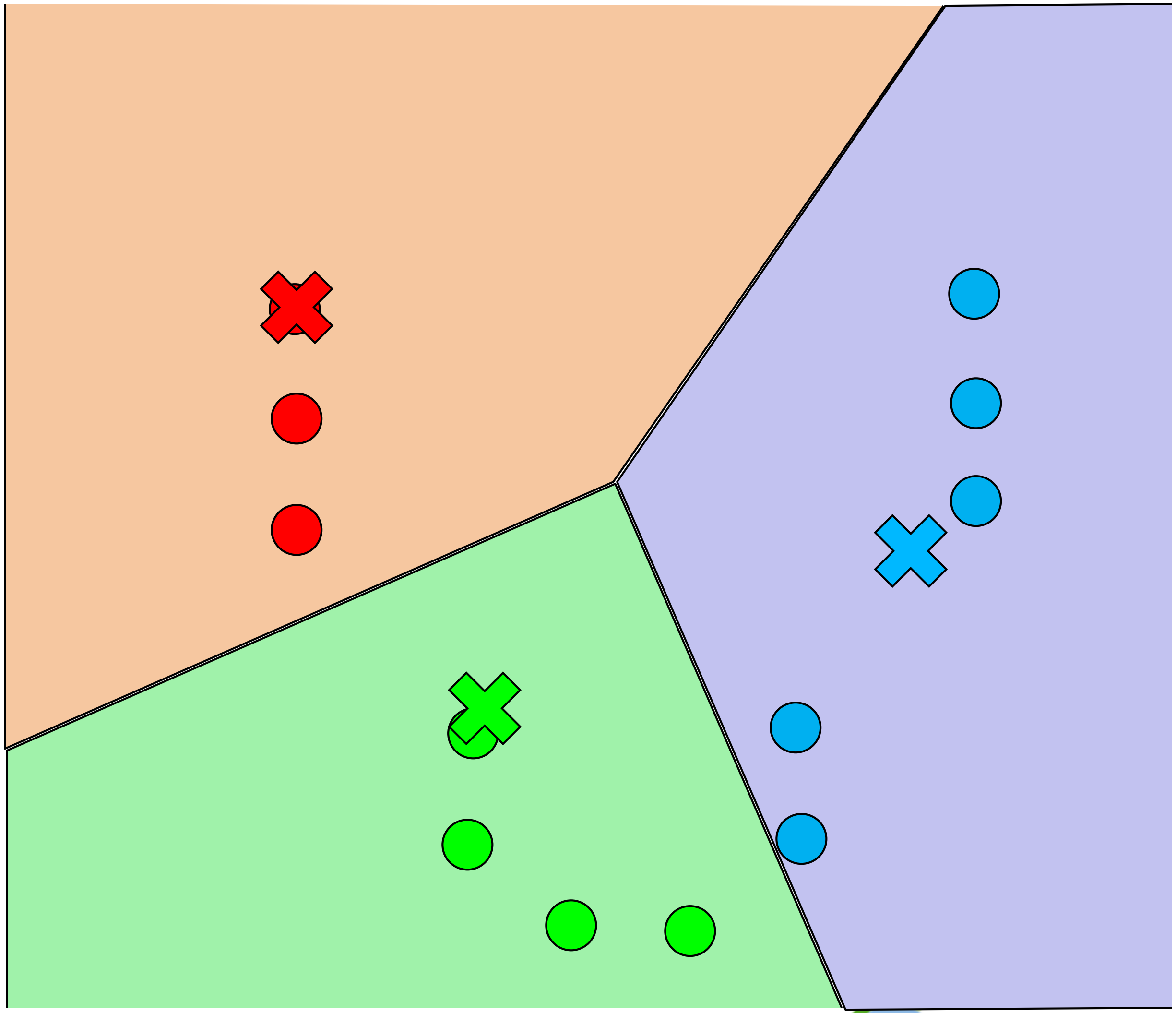
k-means Clustering



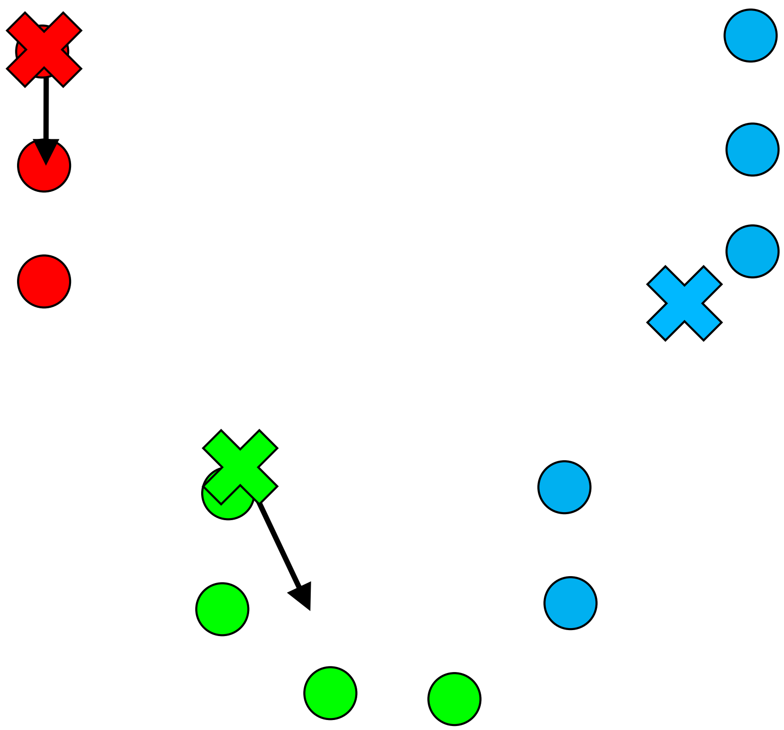
k-means Clustering



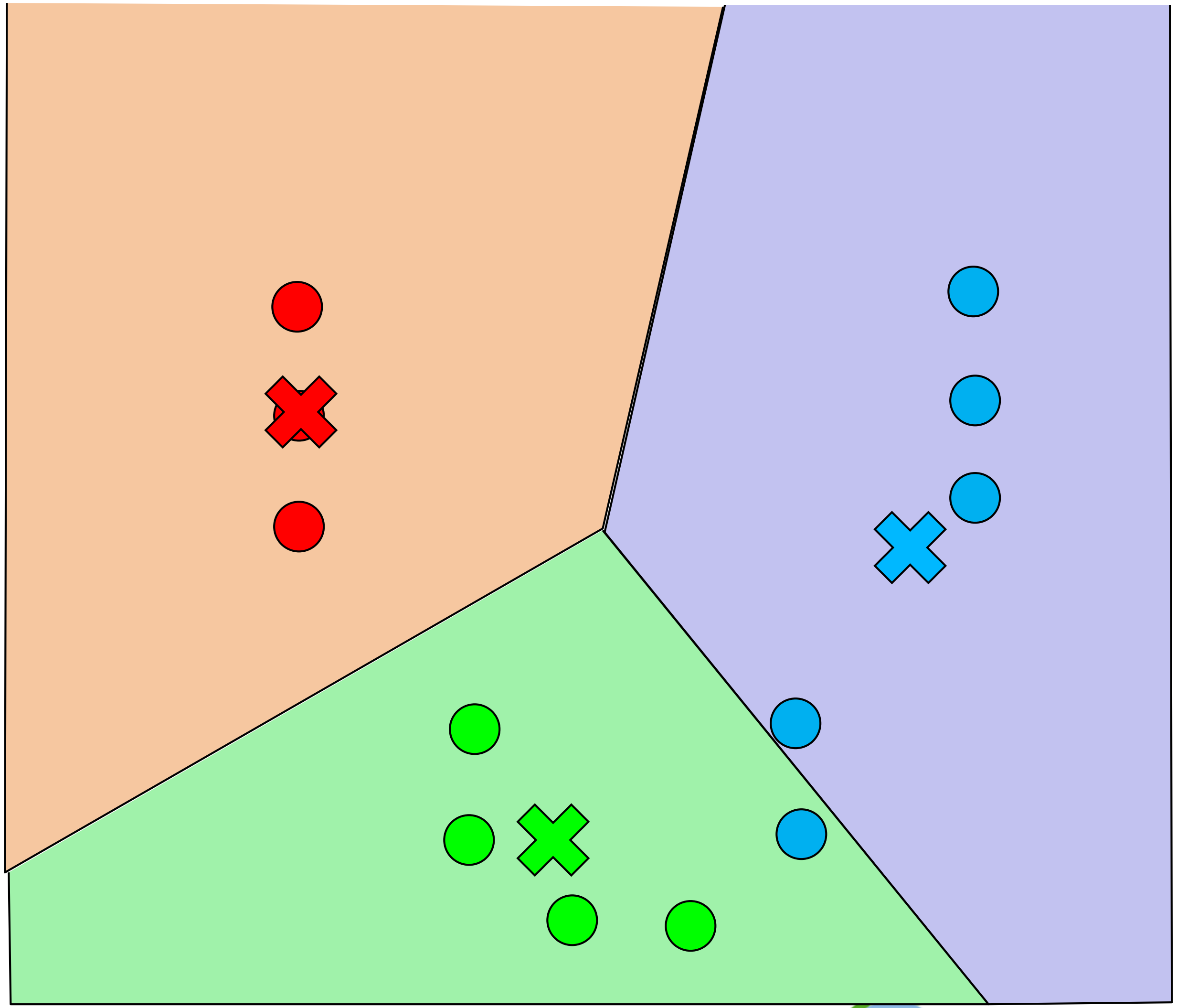
k-means Clustering



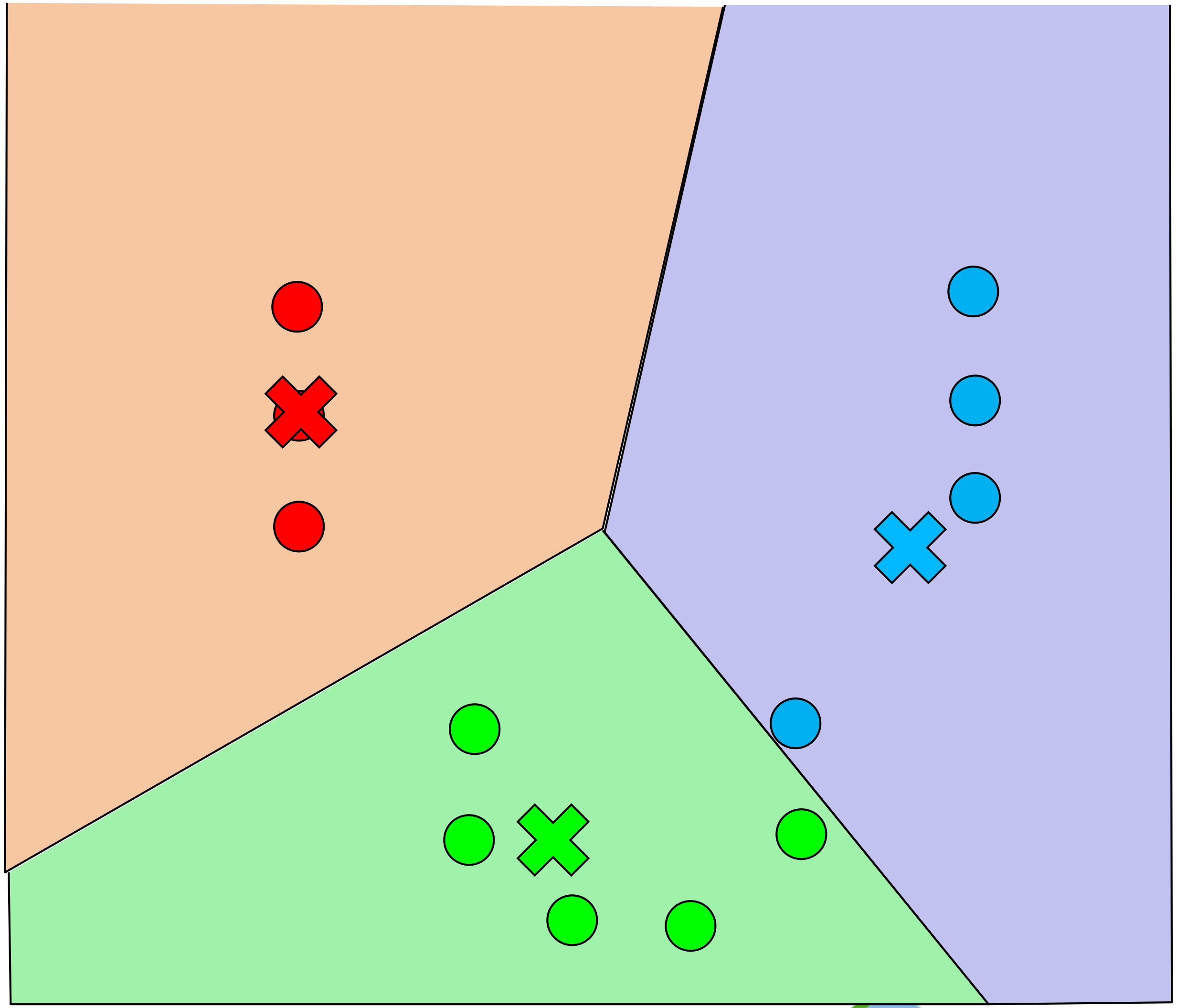
k-means Clustering



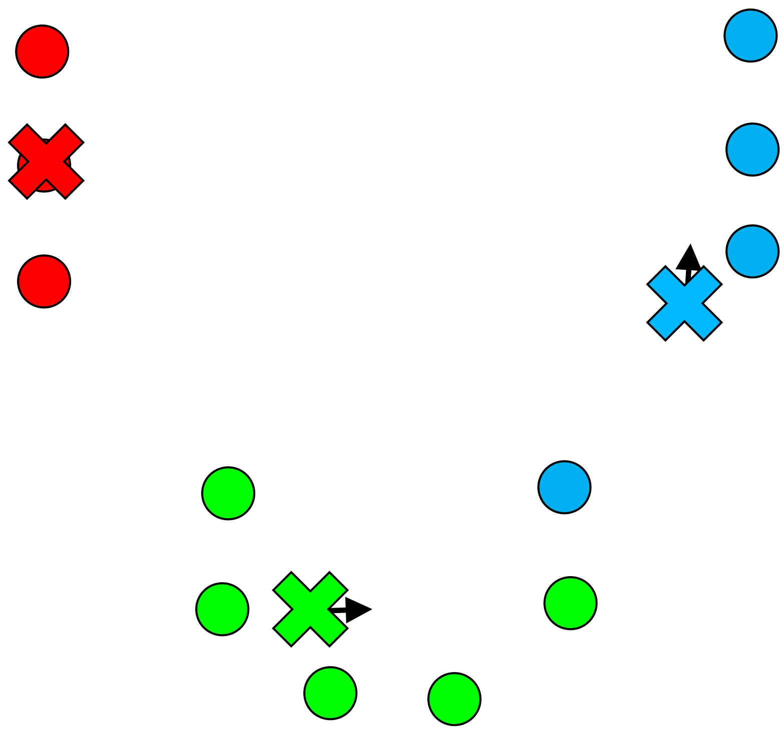
k-means Clustering



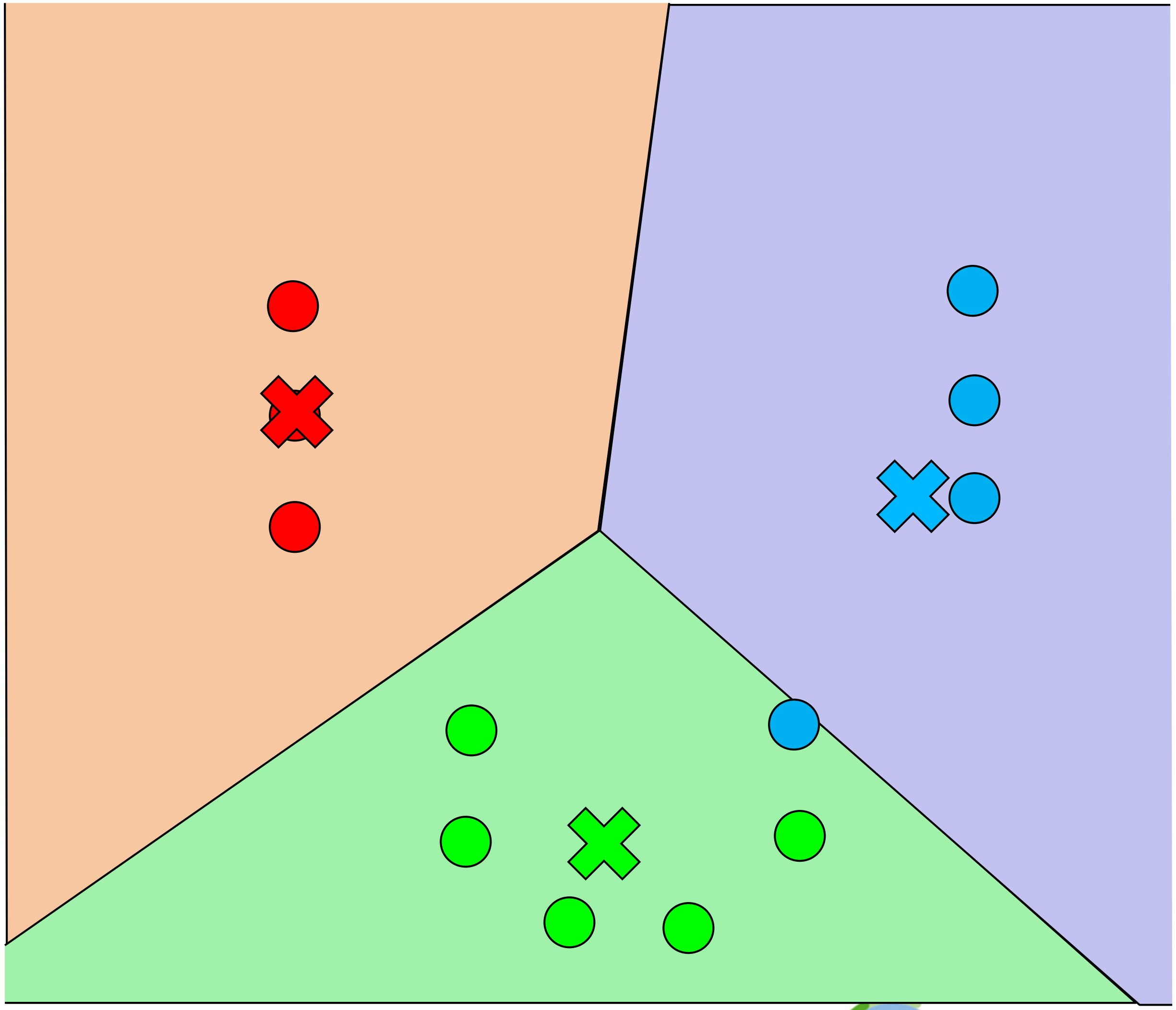
k-means Clustering



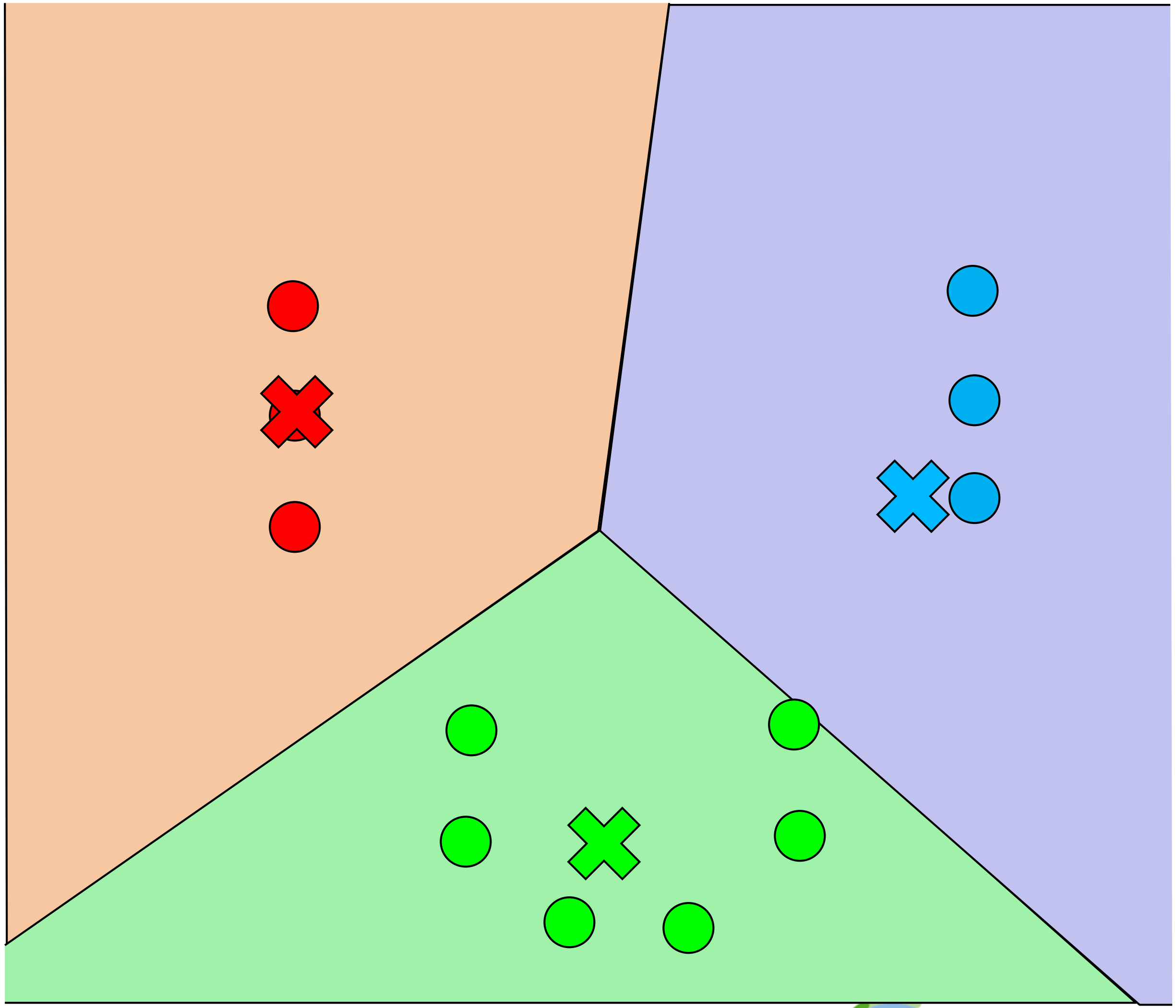
k-means Clustering



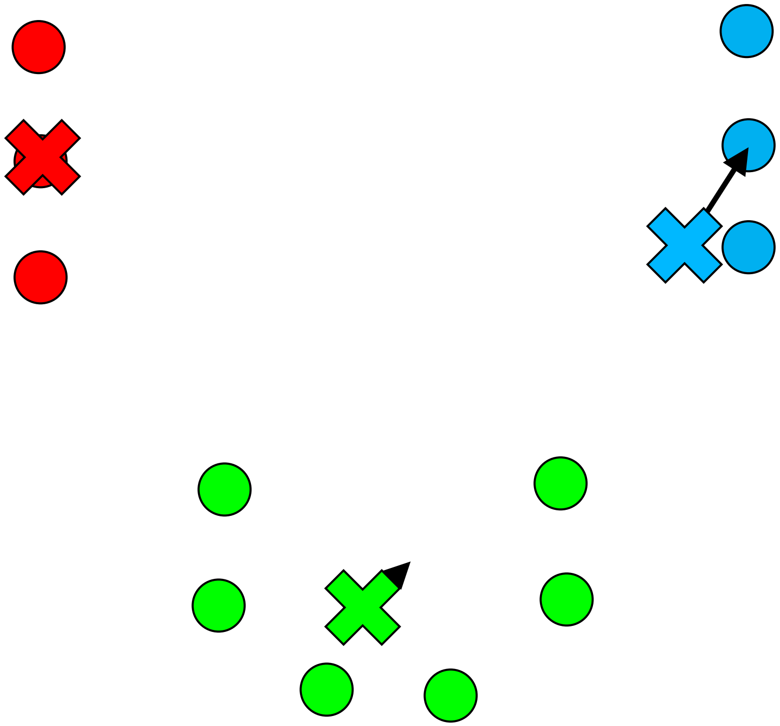
k-means Clustering



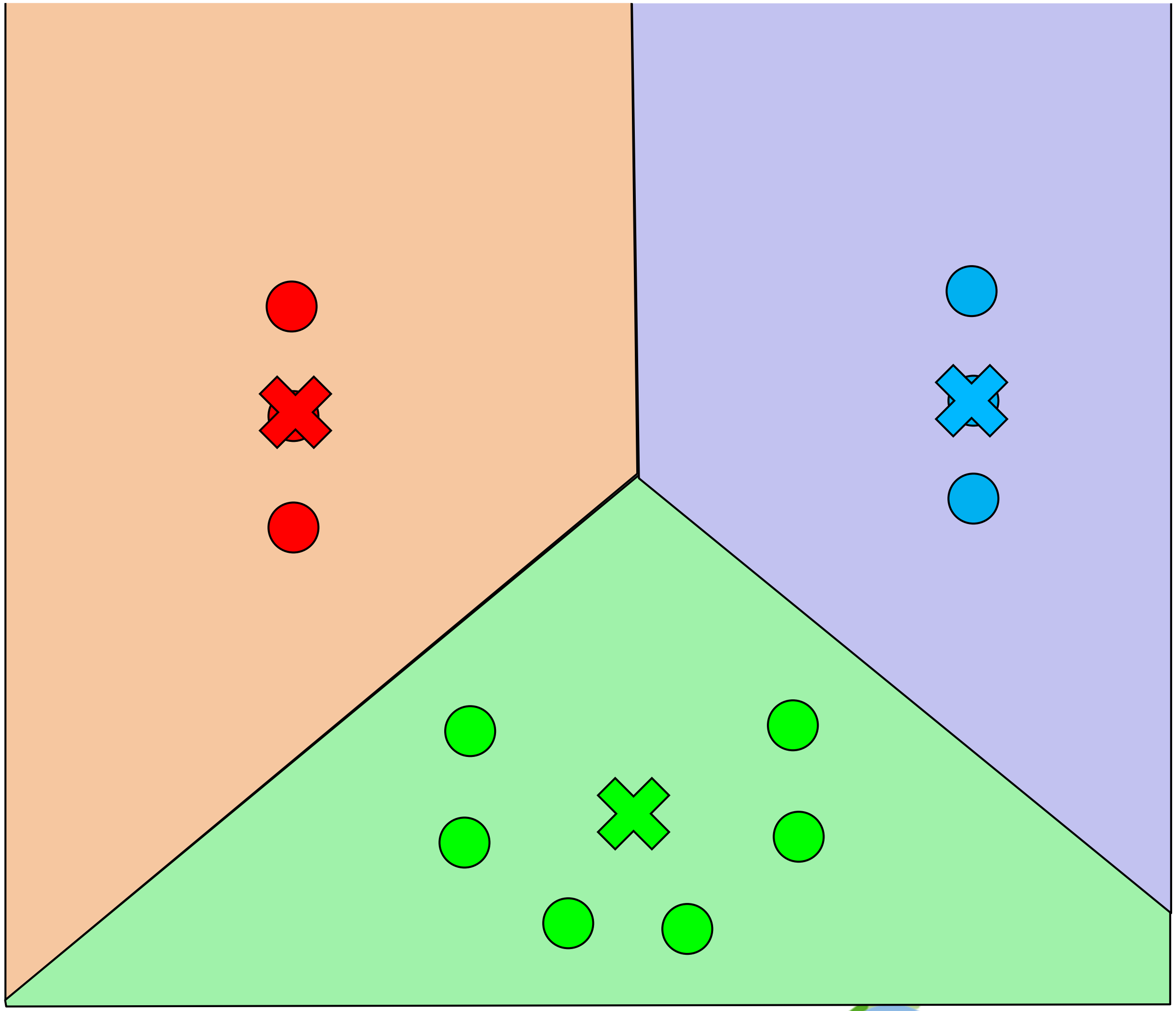
k-means Clustering



k-means Clustering



k-means Clustering



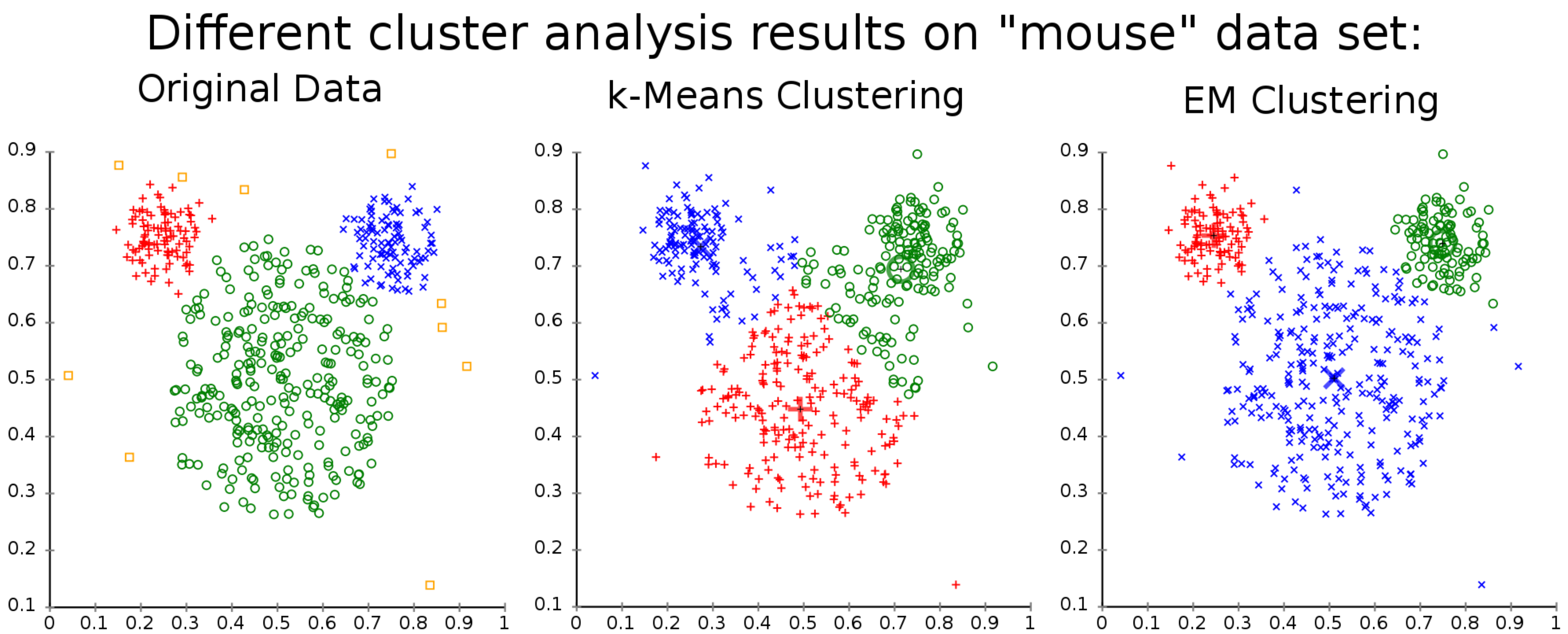
k-means Clustering - Comments

- Advantages:
  - Efficient
  - Always converges to a solution

- Drawbacks:
  - Not necessarily globally optimal solution
  - #clusters k is an input parameter
  - Sensitive to initial clusters
  - Cluster model: data is split halfway between cluster means

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Clustering Results



EM Algorithm

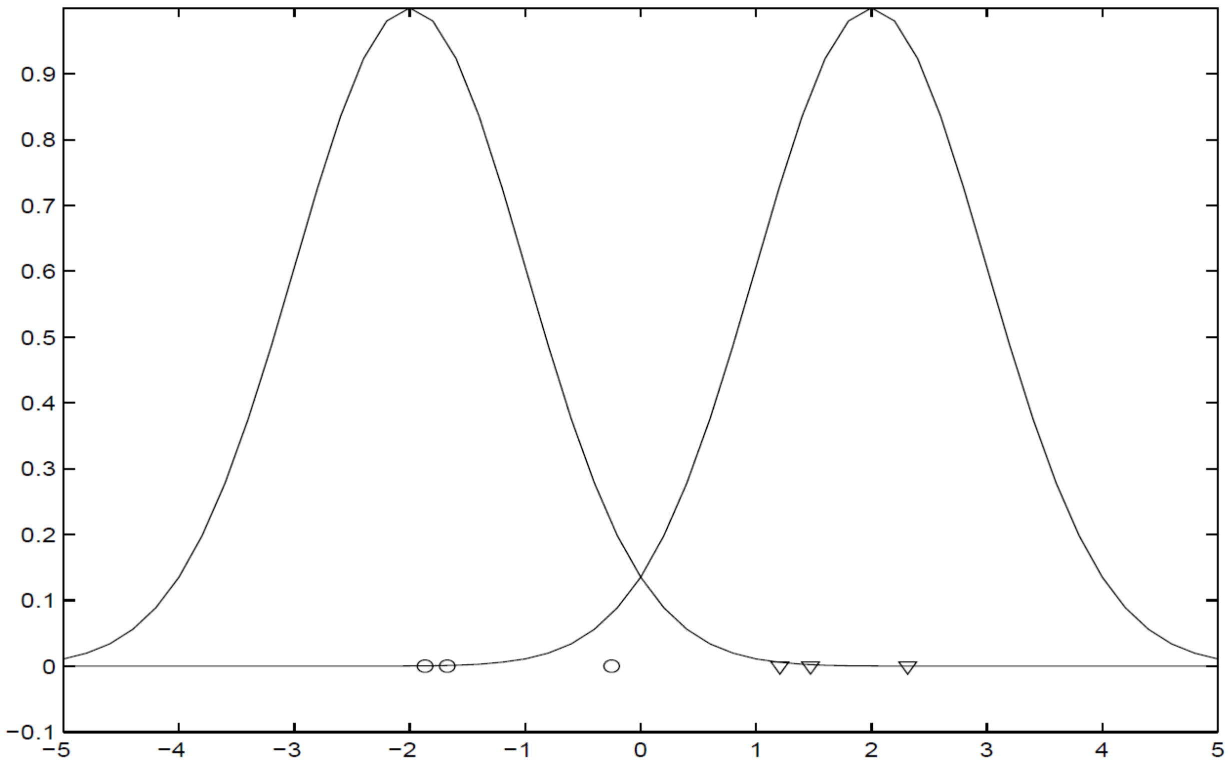
- Expectation Maximization (EM)
- Probabilistic assignments to clusters instead of deterministic assignments
- Multivariate Gaussian distributions instead of means

EM Algorithm

- Given: data samples $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, \quad \mathbf{x}_i \in R^d$
- Assumption: data was generated by k Gaussians
- Goal: Fit Gaussian mixture model (GMM) to data $\mathbf{X}$

  Find $(j = 1, \ldots, k)$
  - means $\boldsymbol{\mu}_j$
  - covariances of the Gaussians $\boldsymbol{\Sigma}_j$
  - probabilities (weights) ω_j that the samples come from the Gaussian j

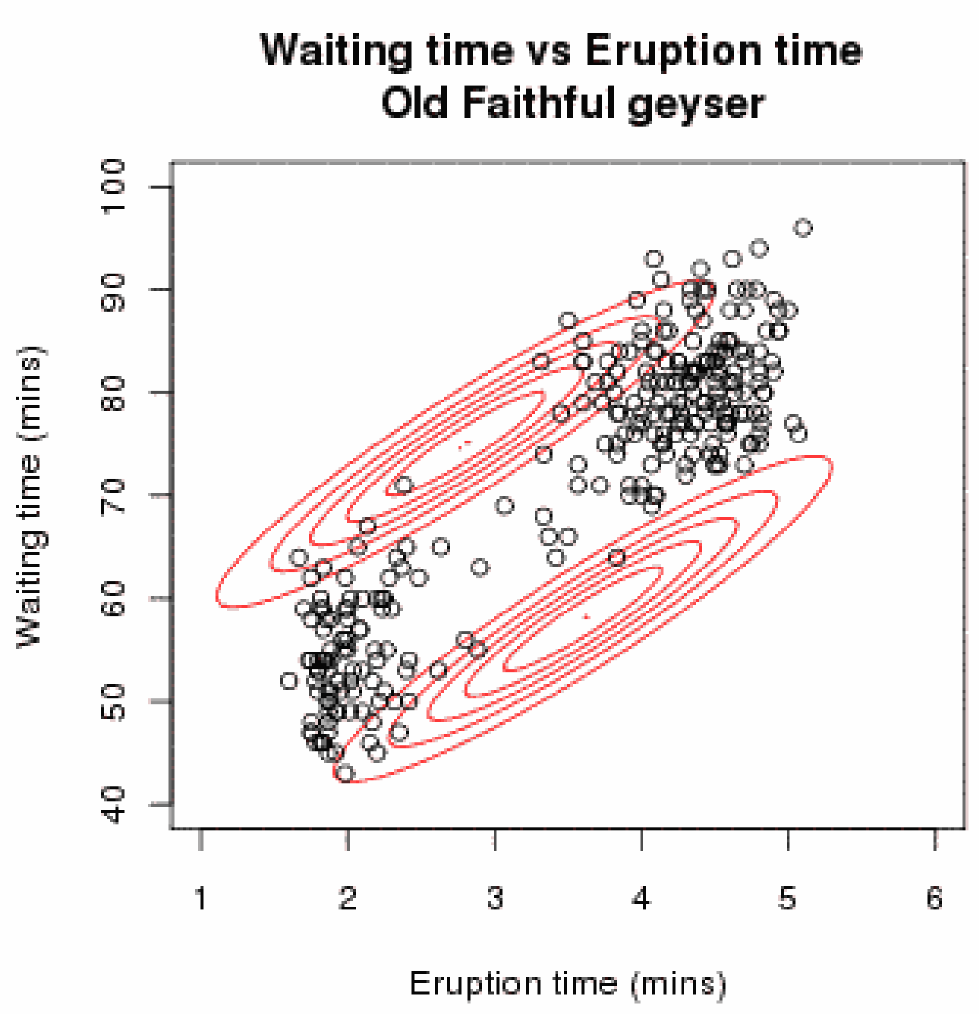
EM Algorithm – Example (1D)



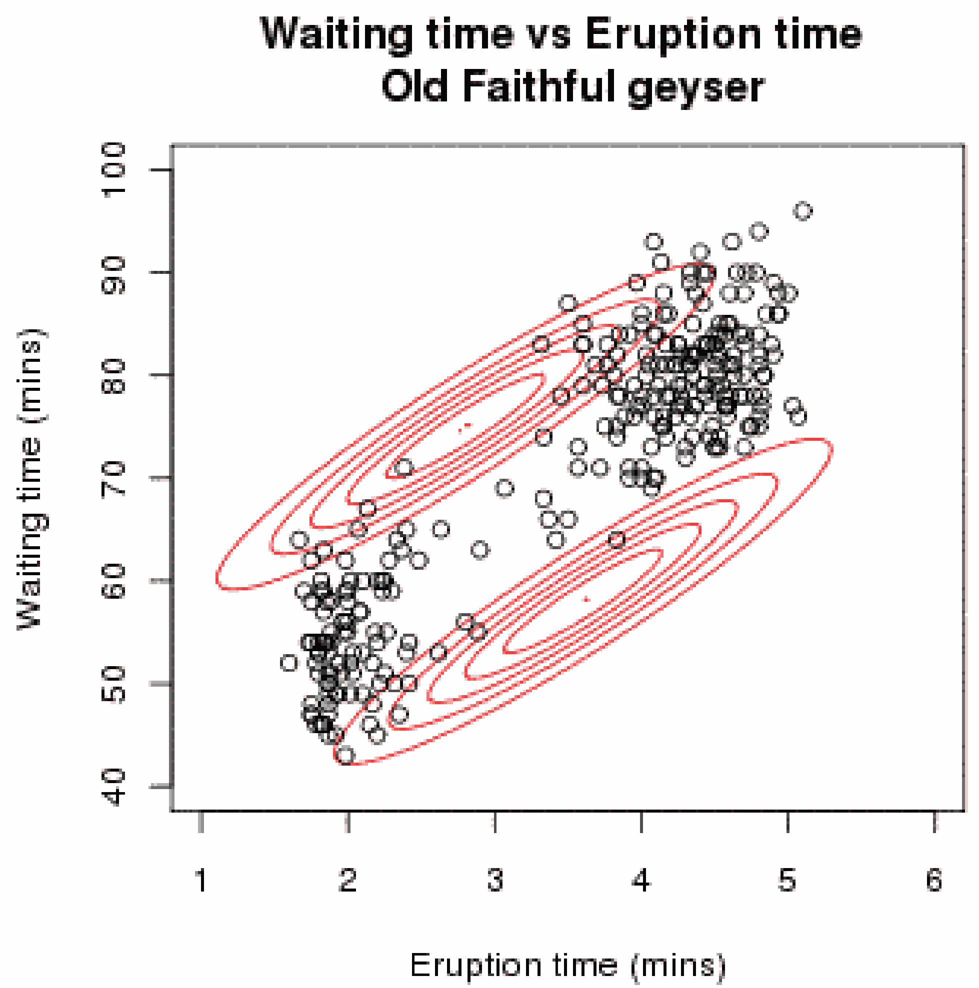
- Three samples drawn from each mixture component
- means: $\mu_1 = -2, \mu_2 = 2$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

EM Algorithm – Example (2D)

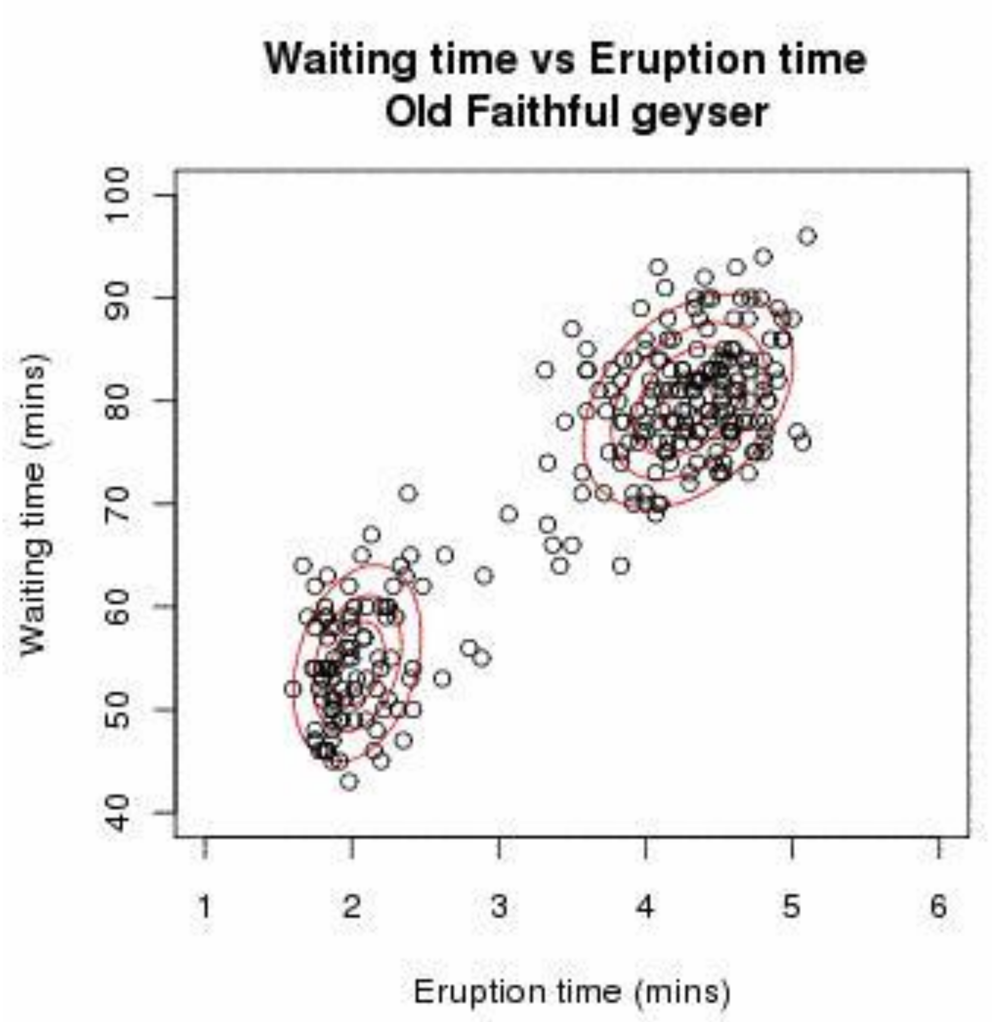


EM Algorithm – Example (2D)



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

EM Algorithm – Example (2D)



EM Algorithm

1. Initialization: Choose initial estimates $\omega_j^0, \boldsymbol{\mu}_j^0, \boldsymbol{\Sigma}_j^0, \quad j = 1, \ldots, k$ and compute the initial log-likelihood

$$L^0 = \frac{1}{n} \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \omega_j^0 \cdot \phi\left(\mathbf{x}_i \middle| \boldsymbol{\mu}_j^0, \boldsymbol{\Sigma}_j^0\right) \right)$$

2. E-step: Compute

$$\gamma_{ij}^m = \frac{\omega_j^m \cdot \phi\left(\mathbf{x}_i \middle| \boldsymbol{\mu}_j^m, \boldsymbol{\Sigma}_j^m\right)}{\sum_{l=1}^{k} \omega_l^m \cdot \phi\left(\mathbf{x}_i \middle| \boldsymbol{\mu}_l^m, \boldsymbol{\Sigma}_l^m\right)}, \quad i = 1, \ldots, n, \; j = 1, \ldots, k$$

and

$$n_j^m = \sum_{i=1}^{n} \gamma_{ij}^m, \quad j = 1, \ldots, k$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

3. M-step: Compute new estimates *(j=1,...,k)*

$$\omega_j^{m+1} = \frac{n_j^m}{n}$$

$$\boldsymbol{\mu}_j^{m+1} = \frac{1}{n_j^m}\sum_{i=1}^{n}\gamma_{ij}^m \mathbf{x}_i$$

$$\boldsymbol{\Sigma}_j^{m+1} = \frac{1}{n_j^m}\sum_{i=1}^{n}\gamma_{ij}^m \left(\mathbf{x}_i - \boldsymbol{\mu}_j^{m+1}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_j^{m+1}\right)^T$$

4. Convergence check: Compute new log-likelihood

$$L^{m+1} = \frac{1}{n}\sum_{i=1}^{n}\log\left(\sum_{j=1}^{k}\omega_j^{m+1}\phi\left(\mathbf{x}_i \middle| \boldsymbol{\mu}_j^{m+1}, \Sigma_j^{m+1}\right)\right)$$

Example (2D)

Ground truth:

– Means: $\mu_1 = \begin{pmatrix} 0 \\ 4 \end{pmatrix}, \mu_2 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$

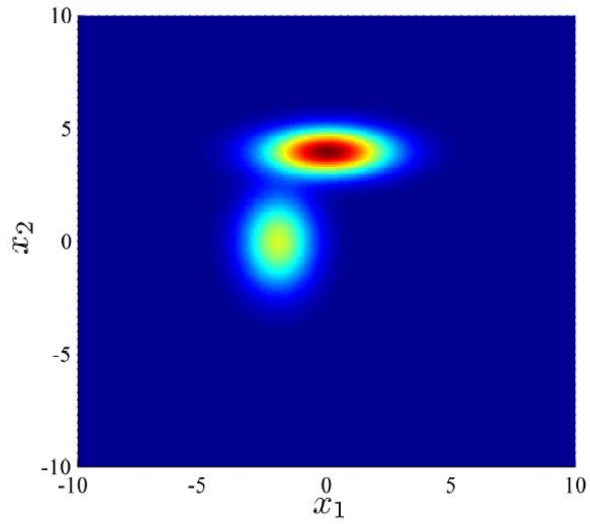
– Covariance matrices:

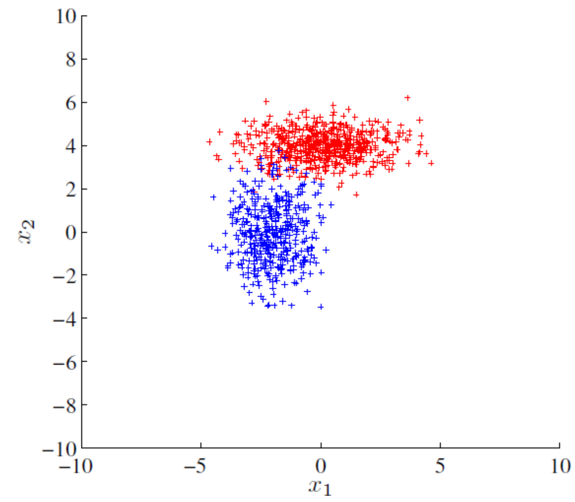
$$\Sigma_1 = \begin{pmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

– Weights:

$$w_1 = 0.6, w_2 = 0.4$$

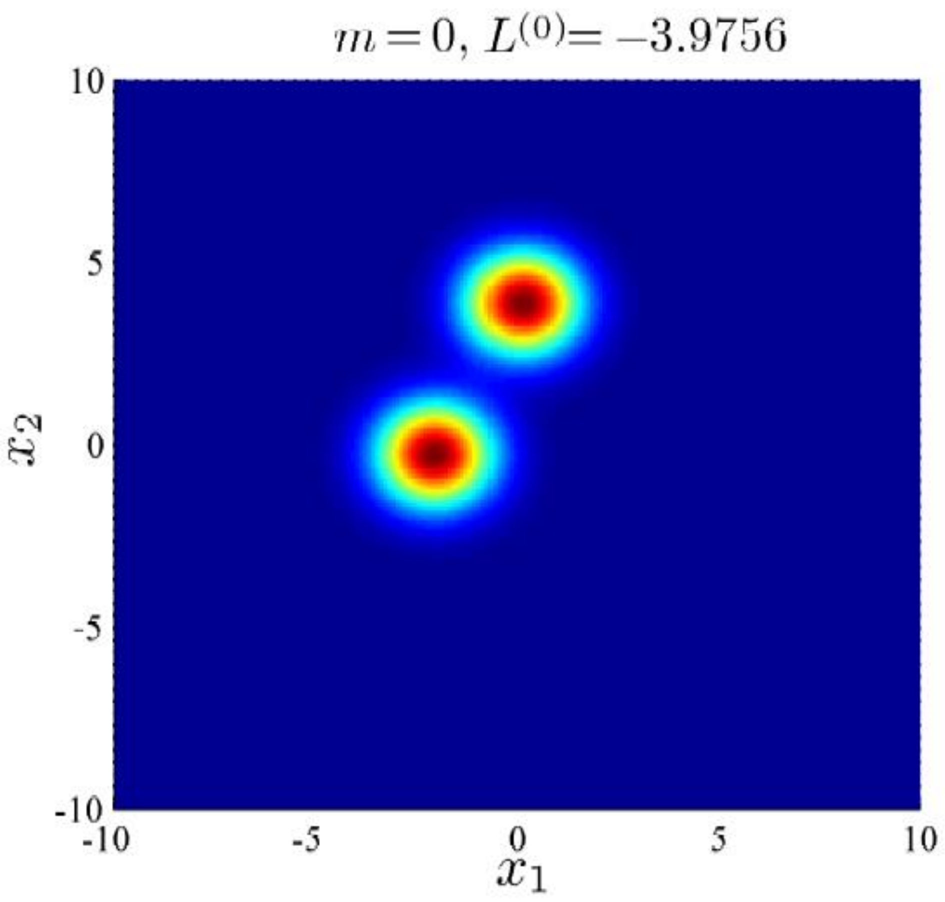
- Input to EM-algorithm:
  1000 samples





Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Initial Estimate



Initial density estimation:

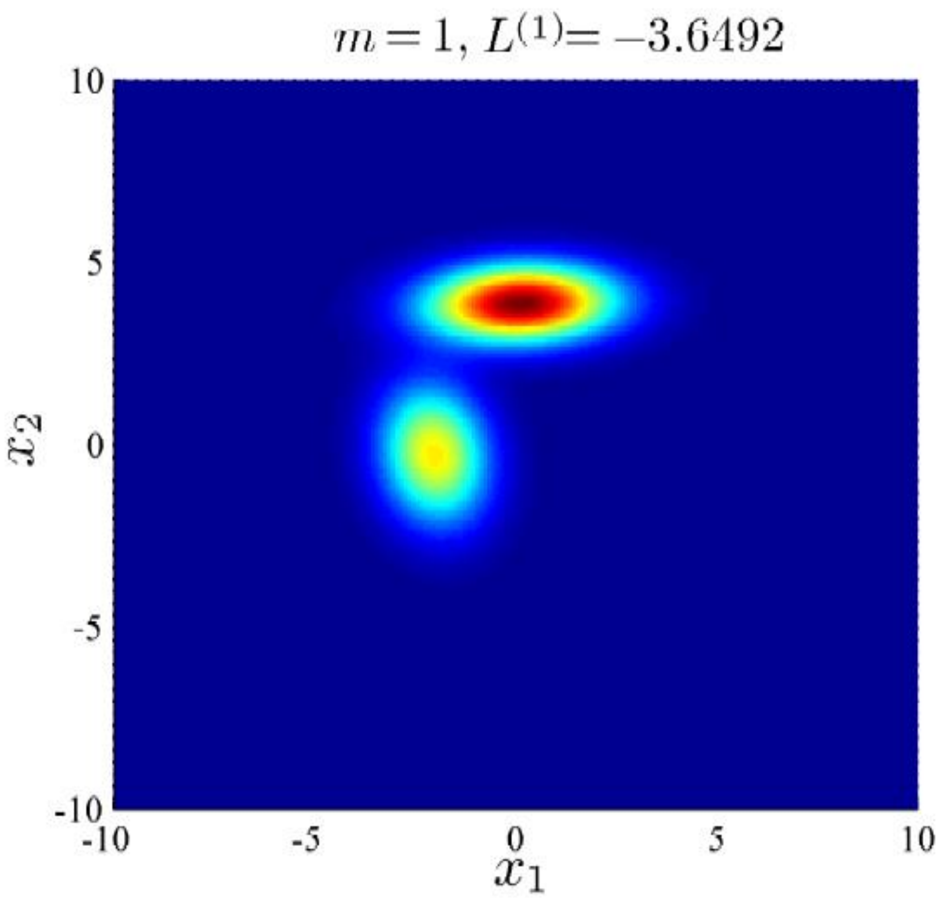
$$\mu_1 = \begin{pmatrix} 0.08 \\ 3.92 \end{pmatrix}, \mu_2 = \begin{pmatrix} -2.07 \\ -0.23 \end{pmatrix}$$

(centroids of k-means result)

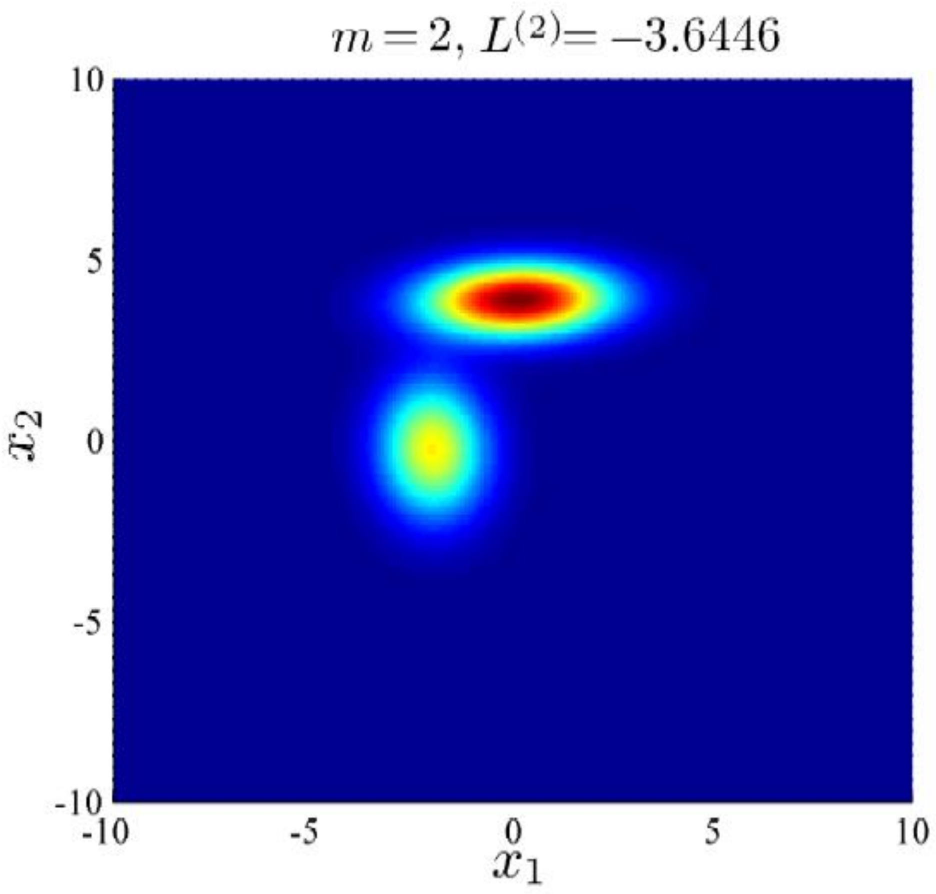
$$w_1 = 0.5, w_2 = 0.5$$

$$\Sigma_1 = \Sigma_2 = I_2$$

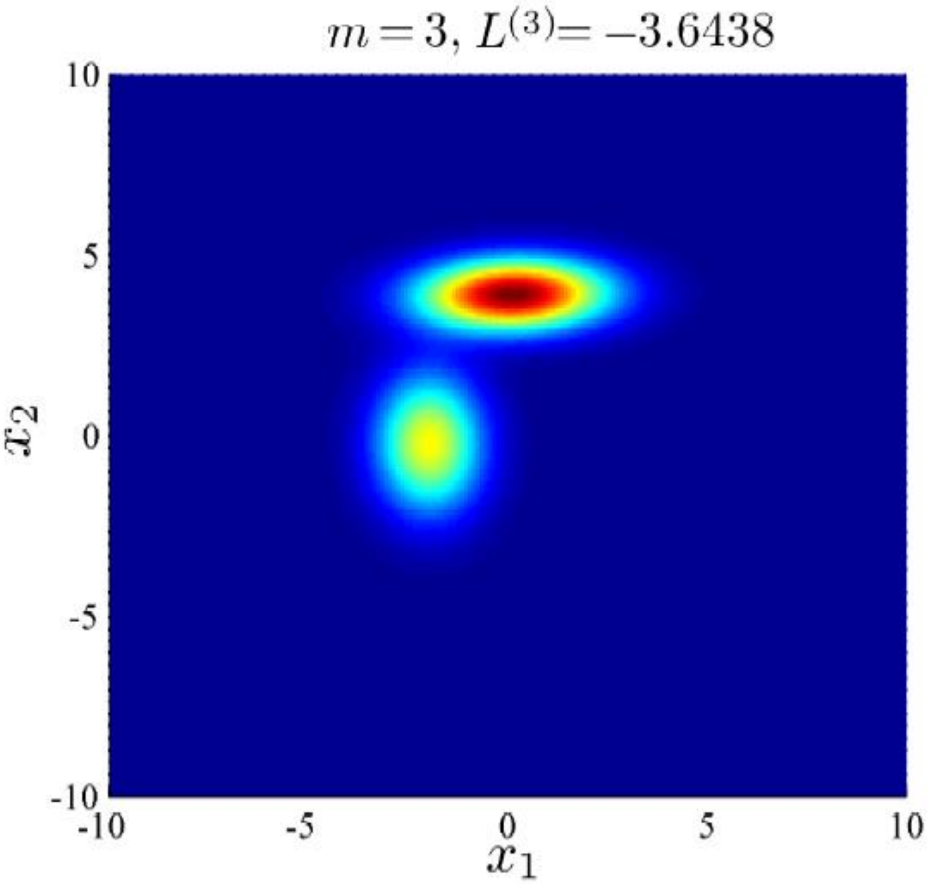
1st Iteration



2nd Iteration



3rd Iteration



Estimates after three iterations:

$$\mu_1 = \begin{pmatrix} 0.08 \\ 3.94 \end{pmatrix}, \mu_2 = \begin{pmatrix} -2.02 \\ -0.17 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 2.75 & 0.06 \\ 0.06 & 0.48 \end{pmatrix},$$

$$\Sigma_2 = \begin{pmatrix} 0.87 & -0.02 \\ -0.01 & 1.79 \end{pmatrix}$$

$$w_1 = 0.59, w_2 = 0.41$$

Mean Shift Clustering

- Non-parametric clustering technique
- No prior knowledge of #clusters
- No constraints on shape of clusters

Mean Shift Clustering - Idea

- Interprete points in feature space as empirical probability density function
- Dense regions in feature space correspond to local maxima of the underlying distribution
- For each sample: run gradient ascent procedure on local estimated density until convergence
- Stationary points = maxima of distribution
- Samples associted with the same stationary point are considered to be in the same cluster

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Mean Shift Clustering

- Given: data samples $\mathbf{x}_1, \cdots, \mathbf{x}_n \quad \mathbf{x}_i \in R^d$
- Multi-variate kernel density estimate with radially symmetric kernel $K(\mathbf{x})$ and window radius h

$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

- The radially symmetric kernel is defined as

$$K(\mathbf{x}) = c_{k,d}\, k\left(\|\mathbf{x}\|^2\right)$$

where $c_{k,d}$ is a normalization constant
- Modes of density function are located at zeros of gradient function $\nabla f(\mathbf{x}) = 0$

Mean Shift Clustering

Gradient of density estimator

$$\nabla f(\mathbf{x}) = \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right) \right] \cdot \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right]$$

where $g(\mathbf{x}) = -k'(\mathbf{x})$ denotes the derivative of the kernel profile $k(\mathbf{x})$

Mean Shift Clustering

Gradient of density estimator

$$\nabla f(\mathbf{x}) = \underbrace{\frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right) \right]}_{\text{proportional to density estimate at } \mathbf{x}} \cdot \underbrace{\left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right]}_{m_h(\mathbf{x})}$$

mean shift vector $m_h(\mathbf{x})$ points toward direction of maximum increase in the density.

Mean Shift Clustering

Mean shift procedure for sample $\mathbf{x}_i$:

1. Compute mean shift vector $\mathbf{m}\left(\mathbf{x}_i^t\right)$
2. Translate density estimation window

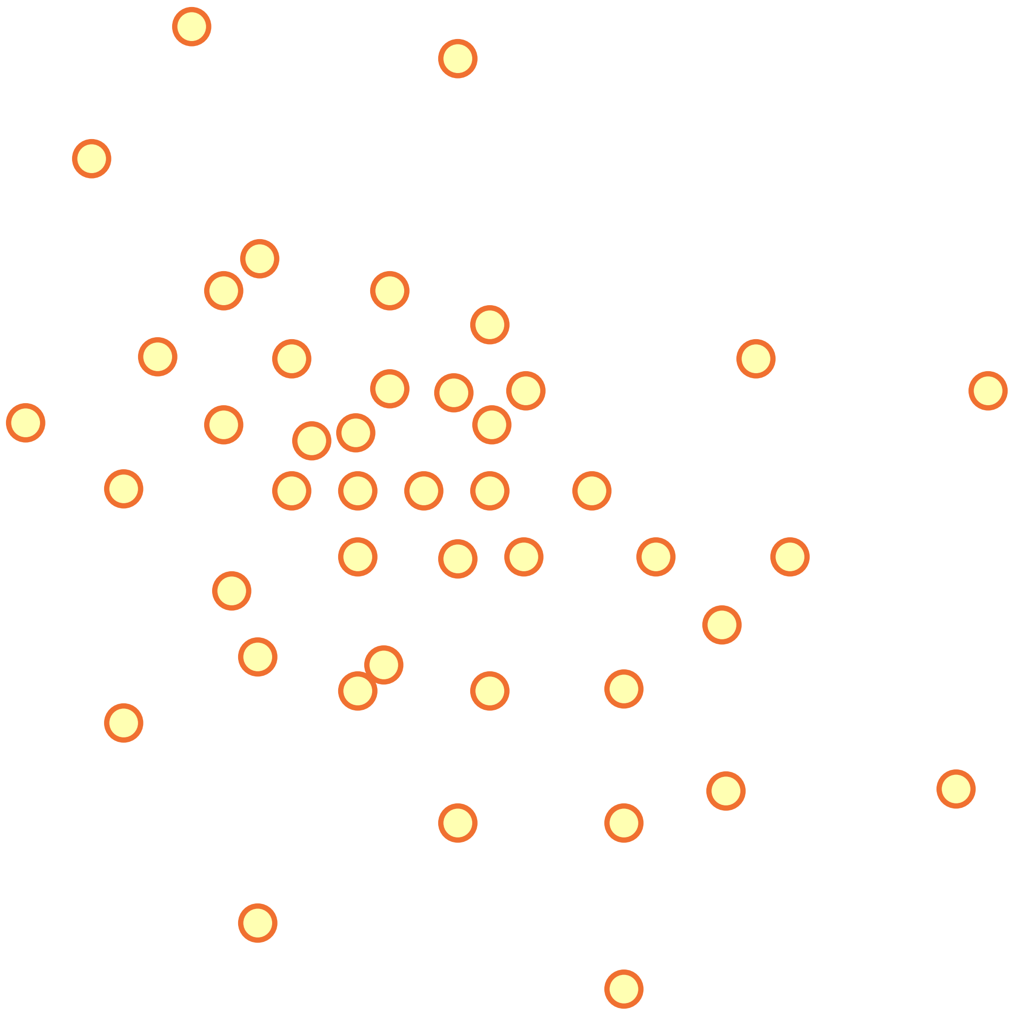
$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + m\left(\mathbf{x}_i^t\right)$$

Iterate 1. and 2. until convergence, i.e.,

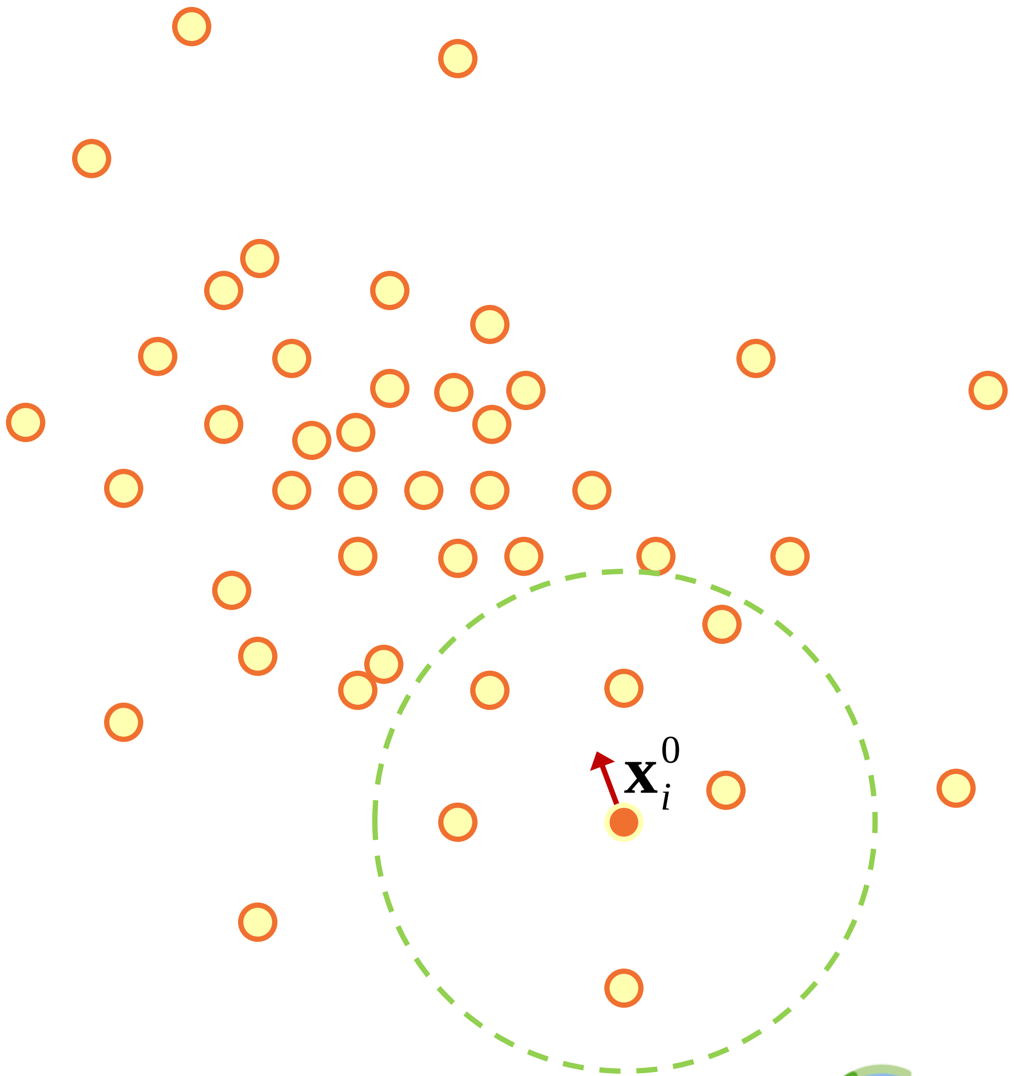
$$\nabla f\left(\mathbf{x}_i\right) = 0$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Mean Shift Clustering

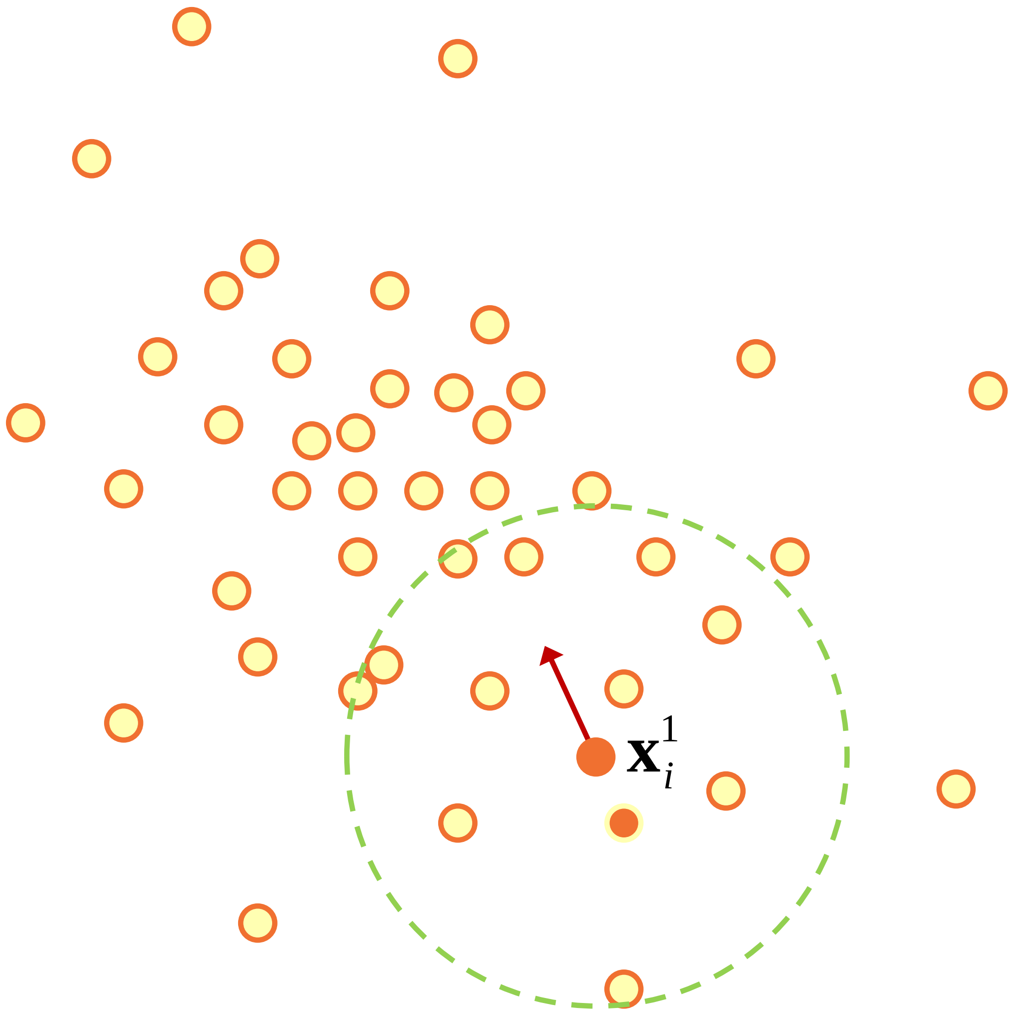


Mean Shift Clustering



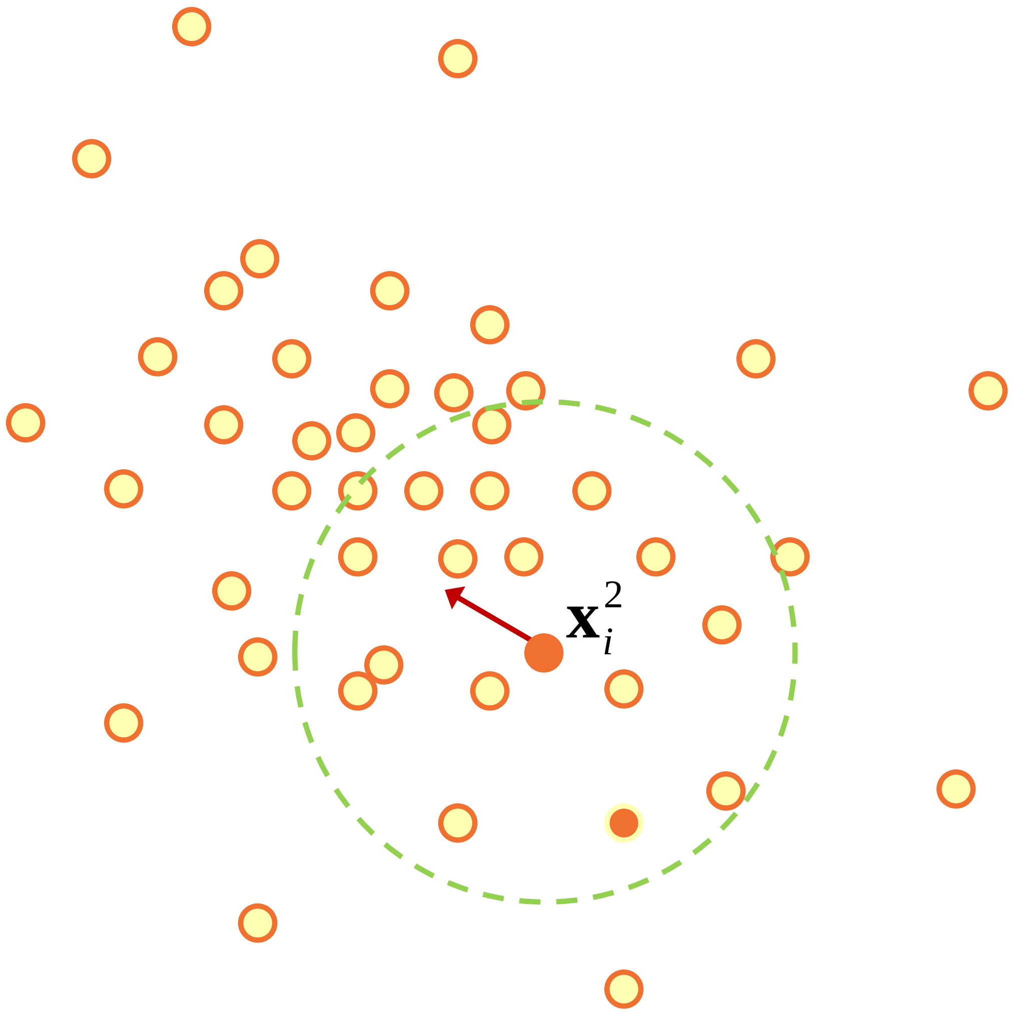
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Mean Shift Clustering

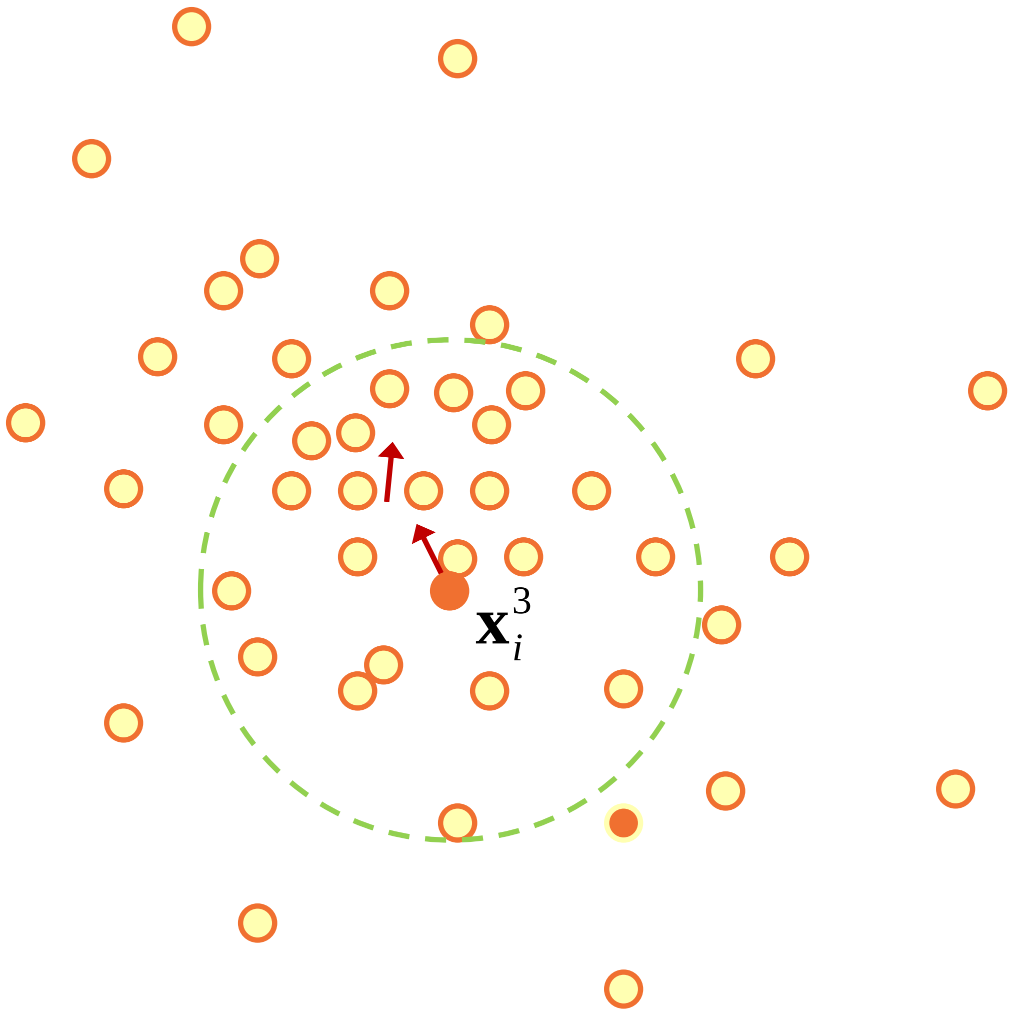


Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Mean Shift Clustering

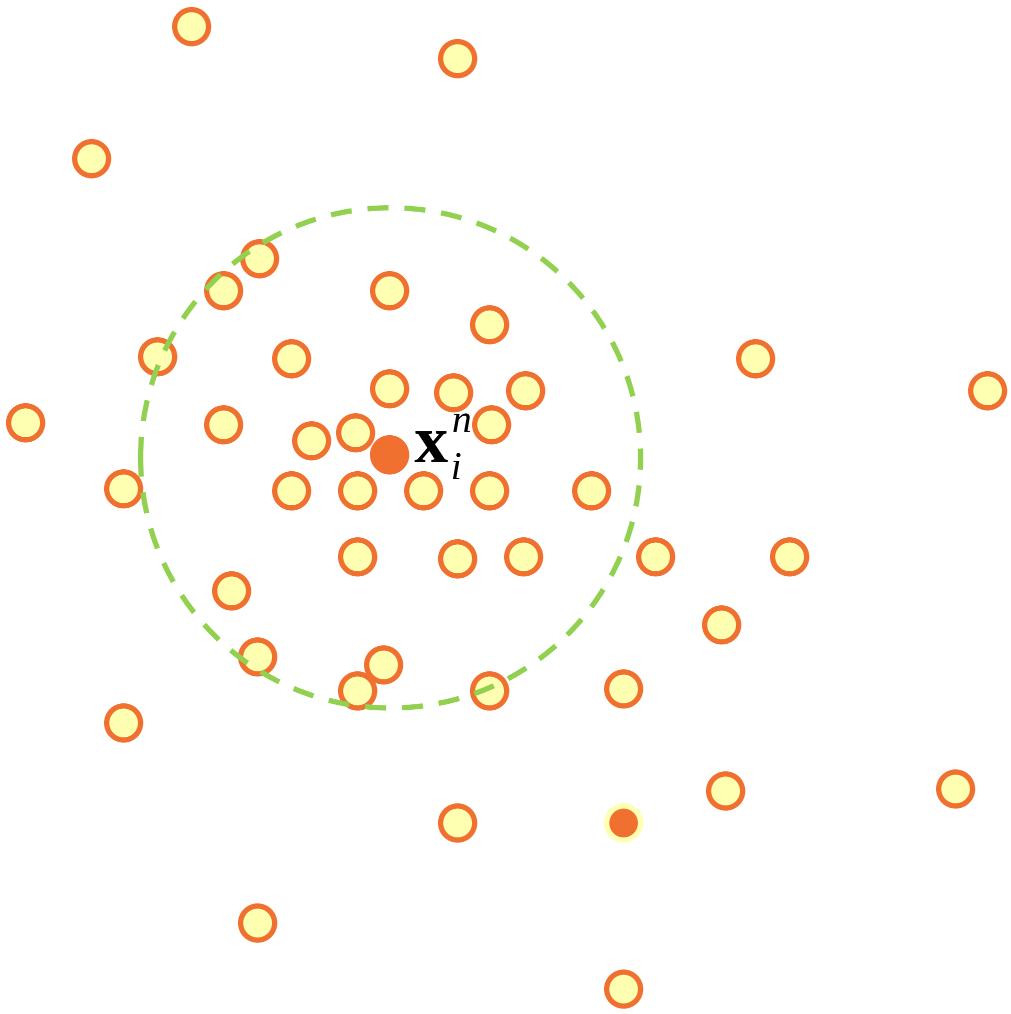


Mean Shift Clustering



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Mean Shift Clustering



Mean Shift - Comments

- Advantages:
  - No prior knowledge of #clusters
  - No constraints on shape of clusters

- Drawbacks:
  - Computationally expensive:
    - Run algorithm for every sample
    - Identification of sample neighborhood requires multi-dimensional range search
  - How to choose the bandwidth parameter h ?

Summary

- Given: n samples in d-dimensional space

$$\mathbf{X} = \left(\mathbf{x}_1, \ldots, \mathbf{x}_n\right) \in R^{d \times n}$$

- Decrease d $\Rightarrow$ dimensionality reduction:
  - PCA
  - MDS
- Decrease n $\Rightarrow$ clustering:
  - k-means
  - EM
  - Mean shift
  - Spectral clustering
  - Hierarchical clustering

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

- Model similarity between data points as graph

- Clustering: Find connected components in graph

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

- Model similarity between data points as graph

- (weighted) Adjacency Matrix W: $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
- Degree Matrix D: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Spectral Clustering

- Graphs:
  - Similarity graph: fully connected, model local neighborhood relations
    - Gaussian kernel similarity function: $w_{i,j} = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}$
  - K-nearest neighbour graph
  - ε-neighbourhood graph

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

- Model similarity between data points as graph

- (weighted) Adjacency Matrix W: $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
- Degree Matrix D: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Graph Laplacian L = D – W: $\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$

Spectral Clustering

- Properties of the Graph Laplacian L:
  - For every vector $f \in \mathbb{R}^n : f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j} (f_i - f_j)^2$
  - L is symmetric and positive semi-definite
  - The smallest eigenvalue of L is 0
    - The corresponding eigenvector is the constant one vector $\mathbf{1}$
  - L has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

- The multiplicity k of the eigenvalue 0 of L equals the number of connected components in the graph
  - Consider k = 1. Assume f is eigenvector with eigenvalue 0:

$$0 = f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j} (f_i - f_j)^2$$

  - The sum only vanishes if all terms $w_{i,j}(f_i - f_j)^2$ vanish
  - If two vertices are connected (their edge weight > 0) $f_i \overset{!}{=} f_j$
  - f needs to be constant for all vertices which can be connected by a path
  - All vertices of a connected component in an undirected graph can be connected by a path:
    - f needs to be constant on the whole connected component

Spectral Clustering

- Laplacian of graph with 1 connected component has one constant vector $\mathbf{1}$ with eigenvalue 0

- For k > 1: Wlog. assume that vertices are ordered according to connected components

$$L = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}$$

- Each L_i is a graph Laplacian of a fully connected graph:
  - Each L_i has one eigenvalue 0 with constant one vector on the i-th connected comp.

- Spectrum of L is given by union of the spectra of L_i

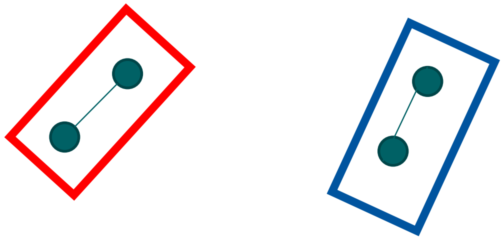
Spectral Clustering

- Graph:

- Graph Laplacian $$\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$
- Eigenvectors for eigenvalues $$\lambda_1 = \lambda_2 = 0 : \begin{pmatrix} 0.71 \\ 0.71 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0.71 \\ 0.71 \end{pmatrix}$$

Spectral Clustering

- Graph:



- Project vertices into subspace spanned by k eigenvectors

$$\begin{pmatrix} 0.71 & 0 \\ 0.71 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{pmatrix}$$

- Projected vertices: $\begin{pmatrix} 0.71 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.71 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.71 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.71 \end{pmatrix}$

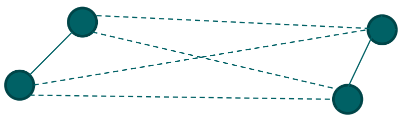
- K-means clustering recovers the connected components
  - Embedding is the same regardless of data ordering

$$\boxed{\begin{pmatrix} 0.71 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.71 \\ 0 \end{pmatrix}}, \boxed{\begin{pmatrix} 0 \\ 0.71 \end{pmatrix}, \begin{pmatrix} 0 \\ 0.71 \end{pmatrix}}$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

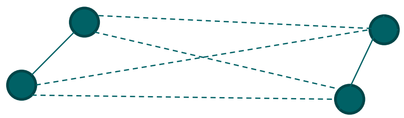
- Similarity Graph:



- W = $$\begin{pmatrix} 0 & 0.9 & 0.1 & 0.1 \\ 0.9 & 0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0 & 0.9 \\ 0.1 & 0.1 & 0.9 & 0 \end{pmatrix}$$

Spectral Clustering

- Similarity Graph:



- L = $\begin{pmatrix} 1.1 & -0.9 & -0.1 & -0.1 \\ -0.9 & 1.1 & -0.1 & -0.1 \\ -0.1 & -0.1 & 1.1 & -0.9 \\ -0.1 & -0.1 & -0.9 & 1.1 \end{pmatrix}$

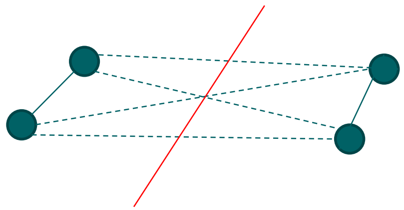
- Eigenvalues : 0, 0.4, 2, 2

- Eigenvectors : $\begin{pmatrix} -0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.5 \\ -0.5 \\ -0.5 \end{pmatrix} \begin{pmatrix} 0.71 \\ -0.71 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -0.71 \\ 0.71 \end{pmatrix}$

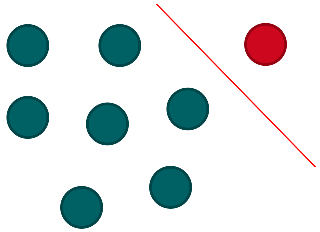
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

- Similarity Graph:



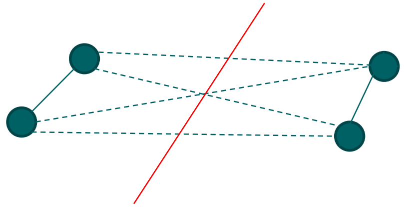
- For fully connected graph we want to find the Min-Cut:
  - Partition graph into 2 sets of vertices such that the weight of edges connecting them is minimal:
    - Vertices in each set should be similar to vertices in the same set, but dissimilar to vertices from the other set
  - Partitions often not balanced: isolated vertices



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

- Similarity Graph:



- For fully connected graph we want to find the Normalized Cut:
  - Partition graph into 2 sets of vertices such that the weight of edges connecting them is minimal
  - Partitions should have similar size

Spectral Clustering

- Min-Cut: minimize $cut(A,B) = \sum_{i\in A, j\in B} w_{ij}$
- Normalized Cut: minimize $Ncut(A,B) = cut(A,B)(\frac{1}{vol(A)} + \frac{1}{vol(B)})$

$$vol(A) = \sum_{i\in A} d_i$$

  - $\frac{1}{vol(A)} + \frac{1}{vol(B)}$ minimal if $vol(A) = vol(B)$

Spectral Clustering

- Reformulate with Graph Laplacian $Ncut(A,B) = cut(A,B)(\frac{1}{vol(A)} + \frac{1}{vol(B)})$
- Construct f: $f_i = \begin{cases} \sqrt{\frac{vol(B)}{vol(A)}}, & \text{if } i \in A \\ -\sqrt{\frac{vol(A)}{vol(B)}}, & \text{if } i \in B \end{cases}$

$$Df^T\mathbf{1} = \sum_{i \in A} d_i \sqrt{\frac{vol(B)}{vol(A)}} - \sum_{j \in B} d_j \sqrt{\frac{vol(A)}{vol(B)}}$$

$$= vol(A)\sqrt{\frac{vol(B)}{vol(A)}} - vol(B)\sqrt{\frac{vol(A)}{vol(B)}} = 0$$

$$f^T D f = vol(V)$$

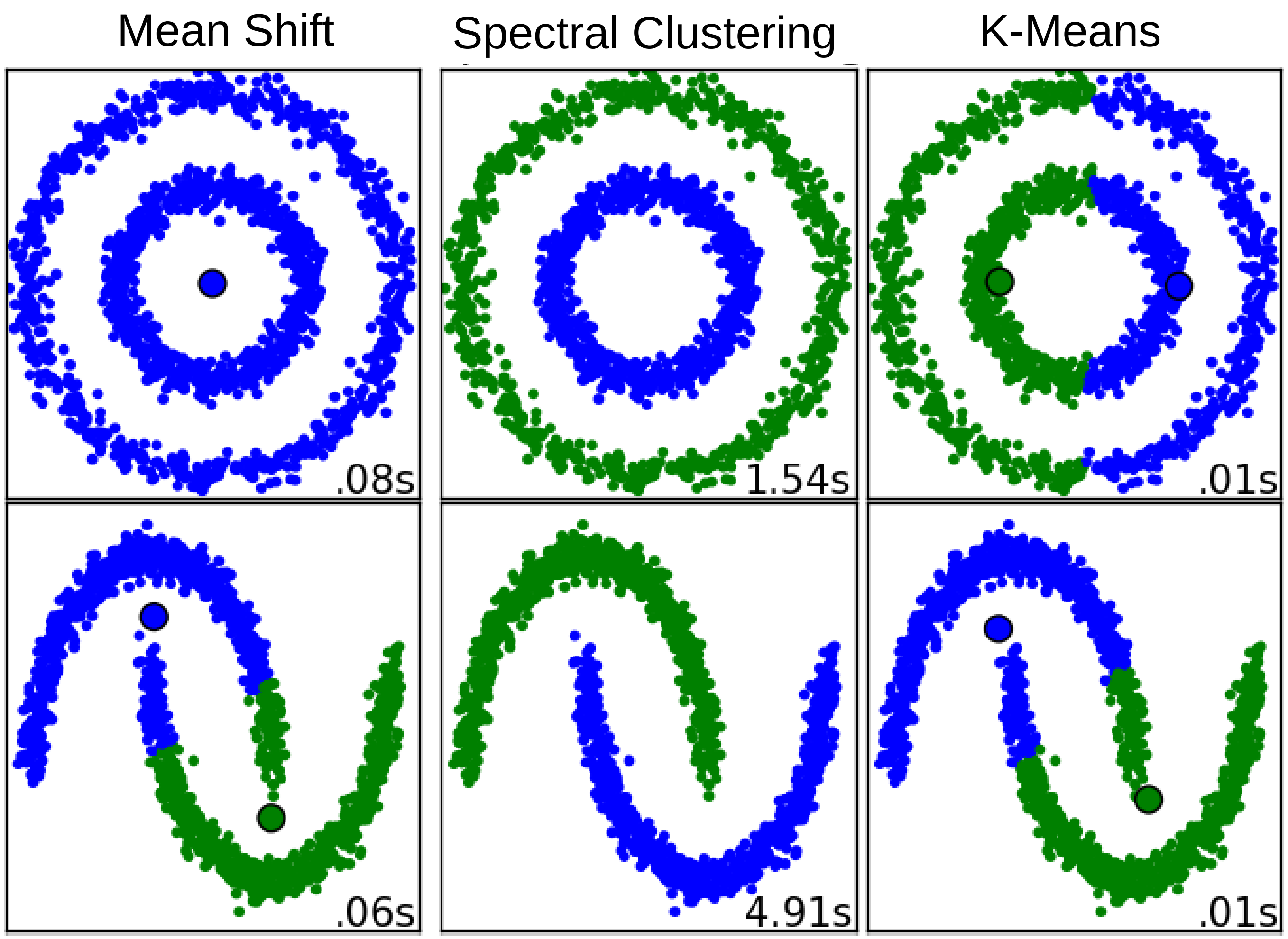
$$f^T L f = vol(V) Ncut(A,B)$$

Spectral Clustering

- Reformulate Ncut: $\frac{f^T L f}{f^T D f} = \frac{vol(V) Ncut(A,B)}{vol(V)}$
- Minimize $\frac{f^T L f}{f^T D f}$ subject to $Df \perp \mathbf{1}$
  - Partition (cluster) assignment by thresholding f at 0
  - NP hard to compute since f is discrete
  - Relax problem by allowing f to take arbitrary real values
    - Solution: second eigenvector of $L' = D^{-1} L$ (normalized Graph Laplacian)
- For k > 2 we can similarly construct indicator vectors like f and relax the problem for minimization:
  - Project the vertices into the subspace spanned by the first k eigenvectors of L'
  - Clustering the embedded vertices yields the solution
- Spectral clustering (with normalized Graph Laplacian) approximates Ncut

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Spectral Clustering

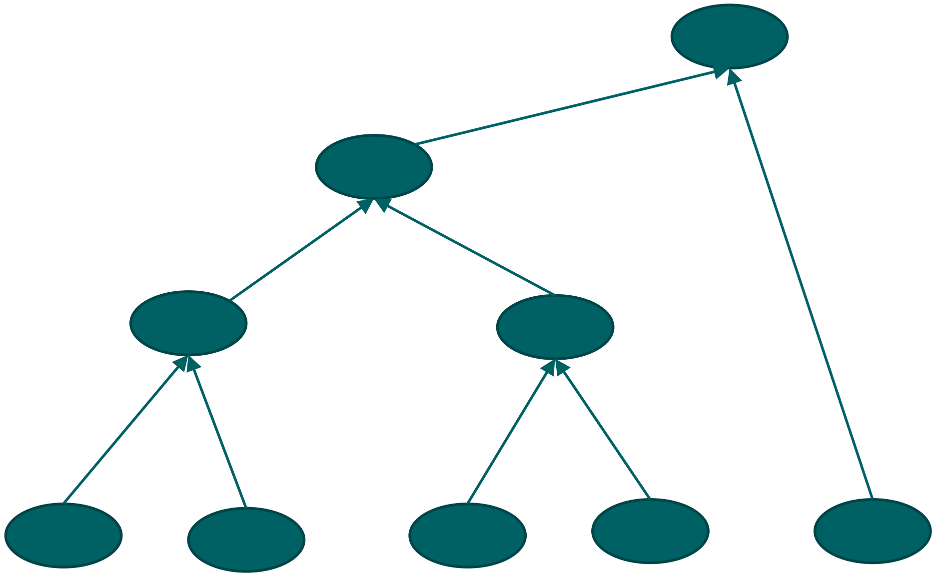


Spectral Clustering

- Summary:
  - Useful for non-convex clustering problems
  - Computation intensive because of eigenvalue computation (for large matrices)
  - Choice of k necessary:
    - A heuristic can be used that tries to find jumps in the eigenvalues (eigengap)
  - Similarity has to be defined for graph construction:
    - Size of Gaussian kernel?
    - Size of neighbourhood?

Hierarchical Clustering

- Bottom up:
  - Each data point is it‘s own cluster
  - Greedily merge clusters according to some criteria



Hierarchical Clustering

- Requirements:
  - Metric: distance between data points $d(x,y)$
  - Linkage: distance between data point sets:
    - Maximum linkage: $l(A,B) = \max d(x,y) : x \in A, y \in B$
    - Average linkage: $l(A,B) = \frac{1}{|A||B|} \sum_{x \in A} \sum_{y \in B} d(x,y)$
    - Ward linkage: $$l(A,B) = \sum_{i \in A \cup B} ||x_i - m_{A \cup B}||^2 - \sum_{i \in A} ||x_i - m_A||^2 - \sum_{i \in B} ||x_i - m_B||^2$$

Hierarchical Clustering

- Algorithm:
  - Start out with a cluster for each data point
  - Merge two clusters that result in the least increase in linkage criteria
  - Repeat until k clusters remain

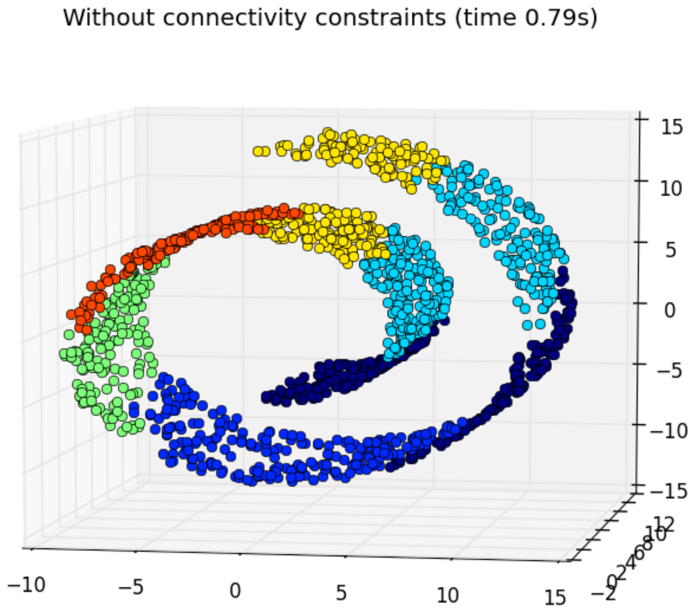
- Maximum linkage:
  - Minimizes maximimal distance of data points in each cluster

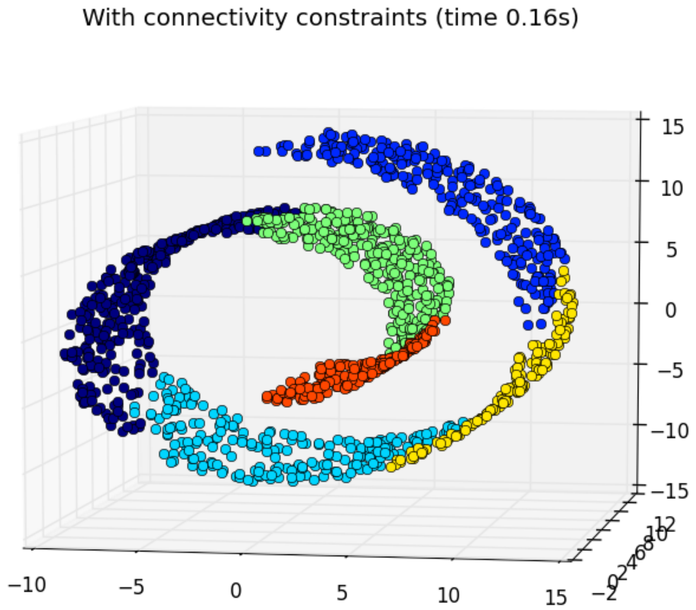
- Average linkage:
  - Minimizes average distance of data points in each cluster

- Ward linkage:
  - Minimizes inter-cluster variance

Hierarchical Clustering

- We can add connectivity constraints that enforce which clusters can be merged





Hierarchical Clustering

- Summary:
  - Flexibel: any pairwise distance can be used
  - Choice of k, distance and linkage necessary
  - Instead of specifying k we can use a heuristic which stops cluster merging if the linkage increases too much
  - Given connectivity constraints hierarchical clustering scales well for large number of data points
  - How do we choose connectivity constraints?
    - K-nearest neighbour graph
    - ε-neighbourhood graph